An Efficient Deadline Constrained Job Scheduling Using Spider Monkey Optimization

Amanpreet Kaur¹, and Mr. Amit Chhabra²

¹Department of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, Punjab, India.
²Assistant Professor, Department of Computer Engineering and Technology, Guru Nanak Dev university, Amritsar, Punjab, India.

Abstract
Finding optimal job scheduling in multi-cluster environments is known to be an NP-hard problem due to heterogeneity of resources and complex mapping of jobs. A common practice is to use basic heuristics to attempt to optimize some performance criteria by treating the jobs in the waiting queue individually. There has been vast research in the field of metaheuristics to achieve optimal results. Swarm intelligence is one of the most promising area that simulate swarming behavior of various creatures like ants, honey bees, fish etc. and the outcomes are actually inspiring. In this paper, a new approach for finding the optimal job schedule is proposed by modeling the foraging behavior of spider monkeys named as (MOSMO), multi-objective Spider Monkey Optimization algorithm. Spider monkeys are fission–fusion social structure-based animals and split themselves based on the scarcity or availability of food from large to smaller groups and vice-versa. The proposed algorithm schedules the jobs considering deadline as constraint to obtain better optimal solution. MOSMO optimizes not only Makespan but also the Flowtime. To compare the proposed metaheuristic, we modified MOSMO without considering deadline and results outperforms.

Keywords: multi-cluster, scheduling, meta-heuristic, nature inspired algorithms, multi-objective optimization.

INTRODUCTION
Job scheduling is considered to be one of the key issues in computing environment. Due to the growth of resources in organizations, the scheduling strategies become great interest in recent years. Job scheduling typically refers to mapping of a set of jobs to a set of resources. This mapping is known to be computationally hard [1]. Considering an objective function such as minimizing the makespan, maximizing resources utilization, minimizing flowtime, maximizing load balancing the quality of resource allocation can be evaluated. The efficiency of scheduler strongly depends on the approach followed to mapping of jobs [1] [2]. In proposed work, we focus is multi-cluster environments. Multi-cluster environments are usually alternative to high performance computing to solve large scale optimization problems. These multi-clusters are made up of several clusters of computers linked by dedicated interconnection networks [2] [3]. In multi-cluster environment, a critical aspect of exploiting the resources is to use co-allocation, that allows the execution of parallel jobs whose computing resource requirements exceed the resources available in a single cluster. Different algorithms could be used to find such mapping, ranging from simple heuristic methods to meta-heuristic methods. In order to enhance the overall performance of multi-cluster system, meta-heuristic approaches are more likely to be preferred [3]. One of these methods is the Genetic Algorithm (GA), a population-based meta-heuristic search method inspired from evolution of living creatures. Unrelated parallel machine scheduling was addressed with resource constraint such as release time, machine requirement etc. Makespan is objective function and two new meta-heuristics GA and artificial immune system was introduced [4]. Job scheduling based on approximation algorithm on heterogenous computational have been introduced. Minimization of the makespan time was the main goal of the proposed scheduler [5]. Three algorithms are presented RR, DRR, dynamic algorithm in order to schedule dynamically with meeting specific reliability and deadline requirement [6]. Dual objective LIB and LCR, quantum inspired GA based load balancing technique for workflow application [7]. We are highly motivated from existing works in the field of scheduling. As scheduling belongs to a category of problems known as NP-hard problem due to large solution space and to find an optimal solution, it takes a long time [8]. There are no such algorithms which results optimal solution within polynomial time to solve these problems. So, it is preferred to find suboptimal solution. Meta-heuristic-based approach evidenced to achieve near optimal solutions for such problems [9]. We propose a new optimization technique that mimics fission–fusion social structure (FFSS) based foraging behavior of spider monkeys. Swarm intelligence-based algorithms able to find near optimal solutions to the large and complex scheduling problems [10]. This approach solves the problem of optimal job scheduling in multi-cluster environment with co-allocation [11] [12] [3]. The jobs are rigid in nature i.e. does not change its requirement throughout the whole schedule. Deadline is considered as constraint to be satisfied [13]. Motivation behind this work is to propose a new meta-heuristic that mimic that swarming behavior of social spider monkey in terms of parallel job scheduling in multi-cluster environment. This work focus on multi-objective functions i.e. makespan, flowtime [14] [15] along with considering deadline constraint.
Multi-clusters have different computing speed and these multi-clusters have same number of computing nodes [16-18]. Jobs are rigid i.e. not malleable.

We are dealing with problem of Scheduling set of Jobs [19] [20].

\{J1, J2, J3, ......Range\} to the available resources satisfying deadline and job requirement in multi-cluster environment. Considering all the jobs arrived at same time.

The rest of the paper is systematized as follows: related work, proposed MOSMO algorithm, experimental evaluation and conclusion

RELATED WORK

Muhanad Tahir Younis et al. (2017) [1], to solve the problem of independent job scheduling in grid computing, a genetic algorithm with a new mutation procedure is presented. To evaluate the proposed method in terms of minimizing the makespan, a known static benchmark is used.

E Gabaldon et al. (2015) [2] proposed algorithm that packages the jobs in the batch to attain better optimization opportunities. A multi-objective function was used to optimize the Makespan of the batches but the Flowtime, thus ensuring a certain level of QoS from the users’ point of view. The proposed meta-heuristic was evaluated with a real workload.

MarinBogeret et al. (2015) [3] presented with approximation algorithm to Schedule jobs with objective to minimize the makespan of a set of \( n \) parallel rigid (and non-preemptive) jobs submitted to \( N \) identical cluster. A new \( 73 \)-approximation is provided running in \( O(\log(n)\log(\text{max} N \cdot n + \log(n))) \).

Mojtaba Afzalirad et al. (2016) [4] addresses an unrelated parallel machine scheduling with resource constrains, sequence-dependent setup times, different release dates, machine eligibility and precedence constrains. To achieve this, a new pure integer mathematical modeling is proposed and ‘makespan’ is considered as the objective function. Two new meta-heuristic algorithms including genetic algorithm (GA) and artificial immune system (AIS) are developed to find optimal or near optimal solutions.

Klaus Jansen et al. (2016) [5] proposed an approximation algorithm with absolute ratio. The goal is to find a schedule for all jobs on the platforms minimizing the maximum completion time (makespan).

Laiping Zhao et al. (2013) [6] examined reliable workflow scheduling with less resource redundancy. They analyzed the reliability of schedule with two definitions accumulated processor reliability and accumulated communication reliability.

Taj Alam et al. (2017) [7] proposed a dual-objective Quantum-inspired Genetic Algorithm based Load Balancing Strategy (QGLBS) for workflow application with the objective of optimizing both Load Imbalance (LIB) and Load Balancing Cost Ratio (LCR). Scheduling the DAG based tasks on heterogeneous distributed system.

Zhao-hong Jia et al. (2016) [8] A meta-heuristic based on Max-Min Ant System (MMAS) is presented for scheduling a set of \( N \) jobs with non-identical job sizes from \( F \) different families on a set of \( M \) parallel batch machines. The objective is to minimize the makespan.

Hadi Mokhtari (2014) [9] introduced a new metaheuristic coined as intelligent water drops (IWD) and adapted to solve a generalized order scheduling problem where with a penalty cost, rejection of received orders is allowed. The objective is to choose the best set of orders having high involvement in manufacture’s benefit

Jagdish Chand Bansal et al. (2014) [10] proposed swarm intelligence approach is named as Spider Monkey Optimization (SMO) algorithm and inspired by intelligent foraging behavior of fission–fusion social structure-based animals.

E. Gabaldon et al. (2017) [11] presented a new hybrid approach of particle swarm optimization and a genetic algorithm to solving scheduling of parallel applications and the resource matching in Federated cluster environments. This minimizes the overall energy consumption and the makespan.

Zhaohong Jia et al. (2017) [12] presented two Ant Colony Optimization-based meta-heuristics, named ACO1 and ACO2. A set of jobs with arbitrary job sizes was used to schedule and release times on a set of P-batch machines with non-identical capacities is considered. Its aim is to minimize the makespan.

Yongsheng Hao et al. (2016) [13] presented an adaptive algorithm to schedule modular non-linear parallel jobs in meteorological Cloud, having a unique parallelism that can only be configured at the very beginning of the execution. This algorithm considered four characteristics of the jobs: the average execution time, the deadlines of jobs, the number of assigned resources, and the overall system loads.

Xiao-Long Zheng et al. (2016) [14] proposed the resource constrained unrelated parallel machine green manufacturing scheduling problem (RCUPMGS) with minimizing the makespan and the total carbon emission. For this, a multi-objective fruit fly optimization algorithm (CMFOA) is proposed.

Mohammad Sajid et al. (2017) [15] projected energy-aware stochastic scheduler to schedule the batch of precedence-constrained jobs on dynamic voltage frequency scaling-enabled processors. Its aim is to optimize the energy consumption and the turnaround time.

César Gómez-Martín et al. (2016) [16] presented Job scheduling algorithm fattened backfilling algorithm provides additional backfilling opportunities, thus more efficient. Shortest jobs move forward if they do not delay the first job of
the queue more than the average waiting time of the already finished jobs. Great improvement in response time and waiting time in most of the cases is recorded.

Hussin M. Alkhashai et al. (2016) [17] introduced two hybrid algorithms based on Particle Swarm Optimization, Best-Fit-PSO (BFPSO) to schedule the tasks is proposed, and PSO-Tabu Search (PSOTS). The performance parameters under consideration are: execution time, cost, and resources utilization.

R. Sundar Rajan (2016) [18] proposes a hybrid algorithm of firefly Max-Min algorithm to schedule of jobs on the cloud. The computing capacity of each machine was in two tiers. The first VM is taken as foreground and the second VM is reserved for background. Objective is minimization of the makespan and flow time of parallel jobs.

Mustafa Muwafak Aloabedy et al. (2015) [19] presented with a high-level hybrid method using ant colony and genetic algorithm to schedule job in grid computing. Results are evaluated using static benchmark problem named as ETC matrix.

Lingfang Zenga, et al (2015) [20] proposed a Security-Aware and Budget- Aware workflow scheduling strategy (SABA), with shorter makespan and providing security services. Results shows under a wide spectrum of workflow applications; the presented scheduling approach is highly effective.

PROPOSED METAHEURISTIC

MOSMO ALGORITHM

1) Initialize Population

Early population of R spider monkeys where each monkey SMr \( (r = 1, 2, ..., R) \) is produced by MOSMO. Here SMr denote the position of rth Spider Monkey (SM) in the population.

2) Local Leader Phase (LLP)

The second phase is Local Leader phase. In this phase, based on the information from the local leader understanding as well as local group members understanding SM update its current location. The fitness value of new location is calculated.

\[
Fitness = \alpha \times (makespan) + (1 - \alpha) \times flowtime \quad \text{...eq}(1)
\]

If the fitness value of the new location is better than previous location, then the SM updates his location with the new one.

3) Global Leader Phase (GLP)

All the SM’s carry their up to date location by Global Leader and local group member’s understanding. Spider monkeys update their locations based on probabilities \( \pi \)’s which are measured using fitness. A better candidate will get more chance to make it better.

\[
Makespan = \text{max} (flowtime) \quad \text{...eq}(2)
\]

\[
Flowtime = \text{Sum} (flowtime) \quad \text{...eq}(3)
\]

Here flowtime = actual execution time + waiting time

The fitness of the newly generated position of the SM’s is calculated and compared with the old one and adopted the better position.

4) Global Leader Learning (GLL) phase

In GLL phase, the location of the global leader is modernized by applying the voracious selection approach in the population. Further, the location of global leader is it is checked whether it is updated or not and in case not updated then, the Global Limit Count is incremented by 1.

5) Local Leader Learning (LLL) phase

The location of the local leader is updated by applying the greedy selection in that group i.e., SM having supreme fitness in that group is chosen as the updated location of the local leader. Furthermore, the updated location is compared with the older one and if the local leader is not updated then the Local Limit Count is incremented by 1.

6) Local Leader Decision (LLD) phase

If any Local Leader location is not updated up to a predefined threshold called Local Leader Limit, then all the members of that group modernize their locations either by random initialization or by using mutual information from Global Leader and Local Leader based on the \( pr \) (perturbation rate).

7) Global Leader Decision (GLD) phase

The location of global leader is monitored and if it is not updated up to a predefined number of iterations that is known as Global Leader Limit, then the global leader divides the population into smaller groups.

Initially, the population is divided into two groups and then three groups and so on till the maximum number of groups (MG) are formed. In case maximum number of groups are formed and even then, the position of global leader is not updated then the global leader combines all the groups to form a single group. As a consequence, the algorithm mimics fusion-fission structure of SMs.
EXPERIMENTAL EVALUATION

I) MOSMO profiling:

The performance of MOSMO is sensitive to different control parameters, such as Size of population / Job range, K parameter. Furthermore, α parameter defines the fitness function which decide how decision process affected by makespan and flowtime.

Deadline is also considered in our proposed work and is one of the key constraint that must be met in order to achieve optimal job scheduling. Deadline is the time span of respective jobs within which these jobs have to be executed. In case Job miss its deadline, then that job never gets executed. So, its crucial responsibility of metaheuristic to schedule jobs in such an order that these jobs executed within deadline altogether with satisfying multi-objective optimization. Deadline is calculated using below mentioned formula:

\[ \text{Deadline} = \text{Arrival time} + K \times \text{Burst time} \quad .... \quad eq(4) \]

Here K is calculated with mentioned formula

\[ K = \text{rand}() \times K_{\text{max}} - K_{\text{min}} + K_{\text{min}} \quad .... \quad eq(5) \]

Where \( K_{\text{max}} = 2 \) or 1.5 or 1,
\( K_{\text{min}} = 0.1 \)

These parameters are fine-tuned by executing MOSMO algorithm, using MATLAB simulation. MOSMO is proposed to solve optimal parallel job scheduling problem in multi-cluster environment along with co-allocation concept. Proposed meta-heuristic algorithm perform scheduling to find optimal solutions in terms of minimization of makespan and flowtime alongside fulfilling deadline constraint.

II) Experiment Setup

In proposed simulation, the Multi-clusters are composed of 4 clusters, each having 32 processor/nodes along with different computation power/ speed.

<table>
<thead>
<tr>
<th>Table [3] Multi-cluster Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLUSTERS</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>CLUSTER 1</td>
</tr>
<tr>
<td>CLUSTER 2</td>
</tr>
<tr>
<td>CLUSTER 3</td>
</tr>
<tr>
<td>CLUSTER 4</td>
</tr>
</tbody>
</table>

The multi-objective fitness is evaluated with initial Job range(population)=300. Jobs are rigid, and arrival time of all jobs is zero.

Generations [1-10] with increment of 1, number of iterations [1-10].
Makespan, Flowtime are multi-objectives that defines the fitness of population.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBS RANGE</td>
<td>300</td>
</tr>
<tr>
<td>GENERATION</td>
<td>10</td>
</tr>
<tr>
<td>ITERATION</td>
<td>15</td>
</tr>
<tr>
<td>K PARAMETER</td>
<td>2</td>
</tr>
<tr>
<td>$\alpha$ PARAMETER</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Initially $\alpha = 0.5$ used to define fitness function. Makespan and flowtime is derived with regard to $\alpha$ parameter. Here $\alpha$ parameter is tuned within [0-1] in order to optimize fitness function [eq (1)].

If $\alpha = 0$ represents flowtime is the dominating factor to evaluate fitness,

If $\alpha = 1$ represents makespan plays key role to evaluating fitness, and rest of the intermediate values represent possible combination of both the parameters i.e. makespan, flowtime. As in Fig [1] [2] depicts higher the makespan, lower the flowtime correspondingly and at certain point, the flowtime start growing.

### III) Experimental assessment:

To evaluate the performance of proposed MOSMO algorithm, we simulate scheduling metaheuristic algorithm in MATLAB.

This simulation is performed to evaluate experimental result in terms of effectiveness of proposal work in multi-cluster setting.

We referred some well-known techniques from the literature, described below

- MOGA, a multi-objective Genetic Algorithm [2]. The findings / result is compared with MOGA, along with Modified version of MOSMO (proposed) meta-heuristic by eliminating deadline from requirement condition, and MOSMO without considering Co-allocation in another version of proposal algorithm.

MOGA [2] treat set of jobs in waiting queue referred as work package. We analyzed behavior of MOSMO (proposed algorithm) by tuning different parameters considered in this work.

Fig [1] [2] depicts makespan and flowtime of MOSMO compared with MOGA (former technique) by varying set of jobs / population ranges from [300-900]. It clearly shows that our MOSMO (considering deadline) perform better by achieving optimization of makespan, flowtime.

![Figure 1](image-url) makespan of proposed and existing metaheuristic
Figure [2] flowtime of proposed and existing metaheuristic

Figure [3] corresponding makespan of proposed and existing metaheuristic when Kmax tuning
Fig [3] [4] represents makespan, flowtime of both proposed (MOSOMO) and existing (MOGA) meta-heuristic when Kmax is tuned from [1, 1.5, 2]. Setting α=0.5, JOBS=500 respectively. We can clearly see MOSMO yield better to minimizes makespan and flowtime in most cases.

Corresponding makespan and flowtime of MOSMO and MOGA is shown in Fig [5] [6].

Here note that α=0.5, JOBS=500, number of nodes / processors varies from [64, 96, 128] in each of 4 clusters. It clearly indicates makespan in case of MOGA is way more than proposed MOSMO this means multi-objective spider monkey algorithm is effective to find optimal scheduling by optimizing makespan as well as flowtime even if after tuning considered parameters. Similarly, we tuned α from [0.2, 0.4, 0.6, 0.8, 1] in order to evaluate makespan and flowtime and these are depicted in Fig [7] [8] respectively. MOSMO outperforms MOGA in effectively minimizing makespan, flowtime in utmost cases.

Fig [9] represents Miss rate of jobs of MOSMO (with co-allocation, deadline) when compared with its modified versions i.e. MOSMO I considering deadline but without co-allocation, MOSMO II without co-allocation and deadline and we are able to reduce miss rate in MOSMO (with co-allocation, deadline). Note that value of Kmax is set to 2 and Jobs varies from [500, 700].

Makespan and Flowtime respective to Jobs [500, 700] is shown in Fig [10]. Note Kmax=2. We compare makespan, flowtime of MOSMO with its other 2 modified variation i.e. MOSMO I, MOSMO II and findings are motivating as MOSMO optimized flowtime, makespan very well.
Figure [6] corresponding flowtime when no. of resources varies

Figure [7] Makespan when α varies in proposed, existing technique
Figure [8] Flowtime when α varies in proposed, existing technique

Figure [9] Miss rate of jobs corresponding modified versions of proposal algorithm i.e. MOSMO
Figure [10] makespan, flowtime respective to jobs=500, 700 of MOSMO and its variations

CONCLUSION

This paper proposes a multi-objective scheduling meta-heuristic for multi-cluster environments. The MOSMO algorithm (proposal) is able to make scheduling and allocation decisions for parallel jobs by considering deadline as key constraint. In this work co-allocation concept is also incorporated to efficiently utilizes the resources by assigning jobs to multiple clusters. The performance of the proposal in terms of makespan and flowtime was evaluated by simulation in MATLAB and compared with multi-objective genetic algorithm (MOGA). The values of different parameters used in proposal were tuned in order to compare and evaluate with existing multi-objective genetic algorithm as well as with modified version of proposed meta-heuristic i.e. (MOSMO without considering deadline). The results show optimization in terms of minimization of makespan, flowtime. Also, the miss rate of jobs in our proposal is less than modified version of proposed algorithm without considering deadline. Our algorithm is able to schedule more jobs to the available resources to fully utilize resources. This clearly indicates proposed MOSMO meta-heuristic based on swarm intelligence is able to achieve near optimal results. These results open new perspective to use further sophisticated multi-objective techniques in increasingly complex environments, that provide decent results in very short period of time.

REFERENCES:


