# Service Flow Management with deadline and budget Constraints using Genetic Algorithm in Heterogeneous Computing

Ahmed A. AbdulHamed, Medhat A. Tawfeek, and Arabi E. keshk
Department of Computer Science
Faculty of Computers and Information
Menoufia University
ahmed_abd1989@yahoo.com, medhattaw@yahoo.com, and arabikeshk@yahoo.com

***Abstract*—The service flow management is one from the most challenges especially in heterogeneous environments which has several and various processors for computing.  Service flow is used to explain services configuration process when service's formation according to the precedence relations of configuration should be considered. Its management should take into account multi-objective constraints. The total execution time should not be completed after the specified time that leading to consider the deadline constraint into account. Also the cost minimization that is a critical issue shouldn't be ignored. Obtaining the optimal management in a sensible time is so hard because there are many candidate with different processing power and price, constraints from the user and the precedence of heterogeneous services. In this paper, the service flow management problem is solved by a genetic algorithm that considers deadline and cost constraints. It focuses on the improvement of execution time to meet the deadline constraint and minimizes the execution cost according to the budget in heterogeneous computing.  The results from the applied experiments proves that the proposed algorithm can be able to minimize total cost, and consolidate the execution time with the deadline constraint. It reach to a near-optimal solution in reasonable time. It outperforms the compared algorithms in the metric of Schedule Length Ratio (SLR), cost, risk ratio, speed up and completion time measurements.**

***Keywords—Heterogeneous Computing; Service Flow; Deadline; Cost; Genetic algorithm***

## I. INTRODUCTION

Heterogeneous computing is a set of techniques that enables various computational capabilities utilization especially for service flow execution [1].  In a general service flow management, it is necessary to assign each service to a suitable candidate and compute the execution order for every service to save the deadline constraint [2].  It assigns many incoming services to available candidates and monitors their execution to optimize some performance measure. The performance measure may be easy as if it takes into account one objective like service flow execution time or cost [3]. It may be more complex if it should consider a mathematical function of various factors such as the cost, deadlines for service flow execution and weighted priorities requirements. All these factors can manipulate and improve the service flow quality of service (QoS) [4].

One of the important factors that are taken into consideration is deadline [5]. Deadline is the final time by which service flow execution must be completed. The service flow management is responsible for different services distribution based deadline to simplify better scheduling decisions [6].

The cost-budget from user should be considered into the management by minimize the cost of service flow execution while continuous meeting the user's deadline level [7].

The Genetic algorithm is a search technique used in computational problems to detect a suitable optimization solutions in a suitable time for hard problems [8]. It contains three main rules or components: selection, crossover and mutation [9]. The selection rule selects the individuals called parents that contribute to establish the population of next generation by a well-defined sequence of operators. Crossover rules collect two parents to form two children for the next population. Mutation rule apply random changes to some individuals [10].

In this paper, the problem service flow management with multi-objective constraints is formulated as a multi-objective optimization problem aiming to simultaneously minimize the cost and improve execution time. After that a Service Flow with Multi-objective based on a Genetic algorithm (SFMG) is proposed. The advantages of SFMG algorithm is that it can achieve best performance in terms of deadline and budget constrained as it has a near optimum solution and doesn't waste time in finding the mathematical calculations to reach best accurate solution.

The rest of this paper is arranged as follows. Section II introduces on related work. Section III discusses the problem formulation. Section IV presents the details of the proposed SFMG algorithm. Section V scans the performance of SFMG with various experiments. Finally, section VI gives a summarization and some future trends.

## II. RELATED WORK

Because the service flow management optimization is very important, many researches have been applied in the literature. Management algorithms are used to map the incoming services of the service flow for the suitable candidates based on some constraints and specific criteria from the user who submit the service flow [3]. The heterogeneity of resource from different candidate and heterogeneity of services lead to make the service flow management as NP-hard problem. The research work in [6], [11] have been proposed to handle only the deadline constraint that leads to increase the consuming cost. From this reason other researches have been proposed to take into account the budget as a main objective [7]. They try to limit the consuming cost based the predefined user budget. Also the Min-Min algorithm in [12] and the Max-Min algorithm in [13] were used to satisfy time and /or cost goals. The PSO algorithm is used to solve the workflow as in [14] [15] [16]. They reduces workflow cost. They consider the data transmission costs and concentrate on decreasing the total computation cost. ACS in [3], [17] has been used to solve service flow in cloud computing with various QoS needs. It consider reliability and response time goals. A novel Greedy- Ant in [18] is introduced to handle the workflow management. This algorithm is used to limit total execution time for the heterogeneous environments. Genetic algorithms as in [8] [9] [10] [19] have been proposed to deal with grid environment for a scheduling problem. These methods work effectively in grid environment but couldn't be directly applied to solve service flow problem in heterogeneous computing. It is reasonable to the user to require multiple objectives such as cost deadline, security, etc. and it is the role of management algorithm to find the scheduling solution which satisfies these objectives. In this paper, a genetic metaheuristic is used to solve the service flow problems taking into account multi-objective constraints in heterogeneous computing.

## III. PROBLEM FORMULATION

This section formulate the problem of service flow management in heterogeneous computing. This formulation is based on the proposed model in [20] for service flow management that include four modules as following:

1. Service Repository module
2. Management module,
3. Broker module
4. Service Level Monitoring (SLM) module.

The flow of service is modelled as *FS* represented by directed acyclic graph (DAG) as in Fig.1.

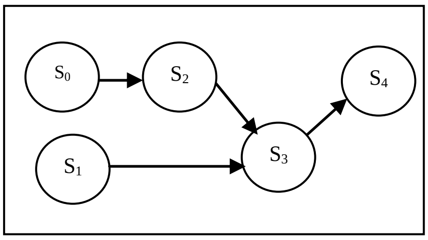

Fig. 1. An example of Service flow

In fig. 1, each vertex represent one service in *FS* and the arcs represents precedence relations between *FS's* services. Every service $S_i$ has domain of Service Candidates *SC*= {$C_1$, $C_{2\ldots}$ $C_m$} such that *SC* is the set of suitable and available candidates and *m* represents they number. Each member from *SC* contain two tuples that represent the cost and execution time for example:

- $C_1$*.cost* represents the cost of candidate $C_1$ for executing the service.
- $C_1$.ET represents the execution time of candidate $C_1$ to execute the service.

The main goal from this service flow management in this paper is to minimize the total service flow execution cost while meeting the deadline constraint. So the service flow management have two objectives. The first as $FS_{Cost}$ stands for service cost objective. The second as $FS_{Deadline}$ stands for service deadline objective.

### A. The cost objective of FS

The cost of service flow $FS_{Cost}$ should be not more than the user specified budget in SLM. The $FS_{Cost}$ is computed by Eq. (1).

$$FS_{Cost} = \sum_{i=1}^{N} S_j^i.Cost \ \le Budget \qquad (1)$$

Where $S_j^i$ is the service $S_i$ that mapped to candidate $C_j$. It is seen from Eq. (1) that for cost optimization, the goal of the managing algorithm is to find a map that satisfies the cost constraint.

### B. The deadline objective of FS

The deadline constraint of *FS* means that the completion time of the last service in *FS* shouldn't be more than the Deadline as in Eq. (2) and Eq. (3). The Deadline is the finishing time that is predefined by the submitter of *FS*.

$$FS_{\text{Completion } TIme} = \sum_{i=1}^{N} S_j^i.ET \times F_i \ \le Deadline \qquad (2)$$

$$F_i = \begin{cases} 1 & if\ \ S_i\ \ is\ on\ the\ critical\ path\ of\ FS \\ 0 & otherwise \end{cases} \qquad (3)$$

If the user who submit the *FS* determines the *Budget and* Deadline, the service flow management algorithm will find a solution that satisfies the constraints in Eq. (1), Eq.(2) and Eq.(3). If the user who submit the *FS* doesn't determine the *Budget or Deadline,* the main goal form the service flow management algorithm will find a solution that:

- Minimizes the value $FS_{Cost}$.
- Minimizes the value of $FS_{\text{Completion } TIme}$

## IV. THE PROPOSED SFMG ALGORITHM FOR SERVICE FLOW

The service flow management for managing the dependent services execution with multi-objective constraints is solved by the genetic concept. It starts with population of chromosomes. It iteratively modifies the current population using genetic operations until stopping condition. Each chromosome is evaluated and ranked by fitness function that holds the meaning of multi-objective constraints. The main steps of the proposed SFMG is introduced in the next subsections.

### *A. Chromosome Representation*

Each chromosome representation gives solution that is represented by a vector of numbers for assigning services process. Each number in this vector is the candidate's reference. Fig. 2 presents the full view of chromosome representation. The upper part of fig. 2 represents the repository that has different sets of candidates for any service. The lower part (chromosome representation) is a vector of selected candidates for the current *FS*. The vector length is equals to the number of services in the current *FS*.

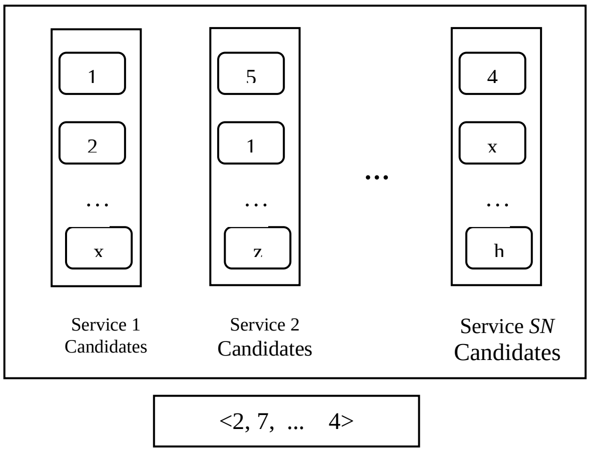

Fig. 2. The view of chromosome representation

### *B. Fitness function*

A fitness function is used to measure the quality of each chromosome in the current generation. The fitness function should collect the heuristic information. It directs the search process from the chromosomes of the proposed SFMG algorithm. It represents the chromosome quality depend on multi-objective constraints.

The fitness function in this paper is a combination of three parts of *FS* execution. Each part measures a single objective to minimize the total cost and deadline in a separated parts as follows.

$$F_{\text{Cost}} = \frac{1}{currentcost} \qquad (4)$$

Where, $F_{Cost}$ is the first part that measures the cost quality and the *currentcost* is cost of the current chromosome solution. Eq. (4) normalizes the value of $F_{Cost}$ to have a value between zero and one and gives a high score for lower chromosome cost.

$$F_{\text{Completion}TIme} = \frac{1}{\text{currentcompletionti}me} \qquad (5)$$

Where, $F_{CompletionTime}$ is the second part that measures the deadline quality and the *currentcompletiontime* is completion time of the current chromosome solution. Eq. (5) normalizes the value of $F_{CompletionTime}$ and gives a high score for chromosome that has a lower completion time.

The fitness function that combine the above three parts is computed by Eq. (6).

$$F = {F_{cost}}^{\alpha} \times {F_{\mathrm{Completion}TIme}}^{\beta} \qquad (6)$$

Where, *F* is the final fitness function the measures the chromosome quality. The α and β are the control parameters the adapt the affection of each objective.

### *C. Selection*

When measuring the quality of each chromosome is computed, selection operation will be proceeded. The proposed method depends on two types of selection.

The first type is the rank selection. During this selection the chromosomes in current population are sorted by the fitness value. The probability that a chromosome will be selected is proportionating to the value of its rank in this sorted list. The rank selection is applied only on selected some chromosomes to be inserted into the next population.

The second type is roulette wheel. During this selection method, the chromosomes are congested into circle of sectors, such that each sector is relative to the fitness quality of each chromosome. The sector who exceed a generated random number, is selected. This selection is iterated until selecting the determined number of chromosomes. The roulette wheel type is applied for selecting parents to create new children by executing crossover and mutation. Using these two type of selection is expected to give an efficient results.

### *D. Crossover*

For creating new chromosomes that is called children, a crossover operation should be applied. It combines a specific parts of the selected chromosomes trying to produce a preferable chromosomes that have high fitness value. The order-based crossover and position-based crossover are used in the proposed SFMG algorithm. During order-based crossover that is applied on two parents, few genes are chosen from one parent randomly, and they order is forced in the second parents and vice versa (The same steps are applied between these two parents conversely to obtain the second offspring) to obtain two children.

The position-based crossover is also applied on two parents to obtain two children but instead of forcing the order of the genes, it forces the position. These two methods of the crossover are used as follows. For the first pair of parents, order-based crossover is applied. For the second pair of parents, position-based crossover is applied and so on until crossover operation is completed. On brief, the order-based crossover is applied on the odd index of pair of parents and the position-based crossover is applied on the even index of pair of parents.

The using of the crossover in this shape give an efficient results that experimentally determined.

### *E. Mutation*

The mutation operation allows in exploring more solutions. It helps a certain child to attain new benefits. The proposed SFMG algorithm uses the exchange mutation method. In this methods, two genes are swapped in the same chromosome randomly. The flow chart of the proposed genetic for service flow management with Multi-objective Constraints is shown in Fig. 3.

## V. IMPLEMENTATION & EXPERIMENTAL RESULTS

The especial simulator based CloudSim in [21] is developed and proposed to test the efficiency of the proposed SFMG algorithm. The configuration of PC is as i5 Core with 4GB RAM with 64-bit Windows 7 operating system. The proposed SFMG is implemented using JAVA programming language on the proposed simulator.

The service flow management algorithms to be compared using the proposed simulator with different experiments include Heterogeneous-Earliest-Finish Time (HEFT) in [18], Greedy Cost (GC) in [22], Ant Colony System (ACS) in [3], L-ACO in [17] and the proposed SFMG algorithm. The HEFT algorithm is a list mapping algorithm which tries to allocate interdependent tasks at minimum execution time on a heterogeneous environment. But here in these paper it used to map interdependent services instead of tasks. The GC approach is to minimize workflow execution cost by assigning tasks to resources of lowest cost. Also it is implemented here for *FS* instead of workflow.

ACS simulates the nature behaviour of real ants. It is proposed to manage service flow taking into account diverse requirements of QoS.

L-ACO uses ant colony optimization to handle deadline-constrained and cost optimization. It constructs an ordered task list and uses the same deadline distribution and service selection methods to build solutions. The comparisons are performed based on the metrics of

- The total cost
- The completion time

- schedule length ratio (SLR) [23]
- speedup [18]

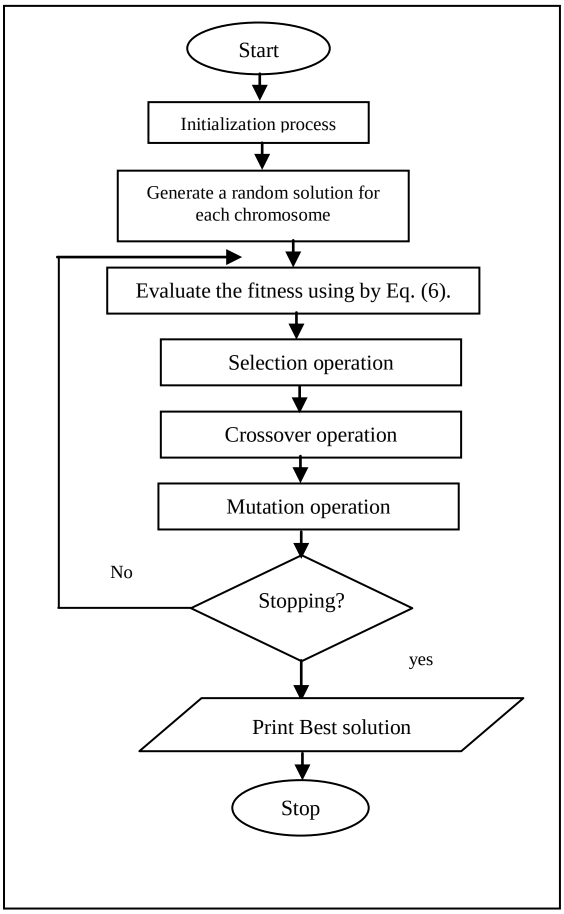

Fig. 3.   Proposed SFMG algorithm

The next experiments include different service flows (*FSs)* that include services from 10 to 100 services. Various service flows (*FSs)* contains different services from 10 to 100 services that are generated randomly by the proposed simulator. In the following experiments the only balanced structure of the service flow is tested.  Table I shows the selected best parameters of the proposed genetic algorithm (GA) that determined experimentally for service flow management.

TABLE I.  PROPOSED GA SELECTED PARAMETERS

| **The parameter** | **Selected Value** |
|---|---|
| *The population size* | *20* |
| *Copying rate* | *0.2* |
| *crossover rate* | *0.8* |
| *mutation rate* | *.05* |
| $T_{max}$ | *100* |
| $\alpha$ | *1* |
| $\beta$ | *1* |

The proposed SFMG algorithm has a population that includes 20 chromosomes. The copying rate represents the rank selection rate that is applied only on selected chromosomes to be inserted into the next population. It is equals to 20% from the total population size. The crossover rate is equals to 80% from the total population size and the mutation rate is applied only on 5% from the total population size. The proposed SFMG algorithm depends on a fixed number of iterations as stopping criteria that is set to 100.  The cost and the capability of any candidate is assumed to be known for any service. They are assigned with random value in this paper by the proposed simulator. The following experiments is executed more than ten times to get the average values. The average consumed cost of the HEFT, GC, ACS, L-ACO and the proposed SFMG algorithms is presented in Fig. 4. The proposed SFMG algorithm has a less average consumed cost than HEFT, GC, ACS, L-ACO algorithms.

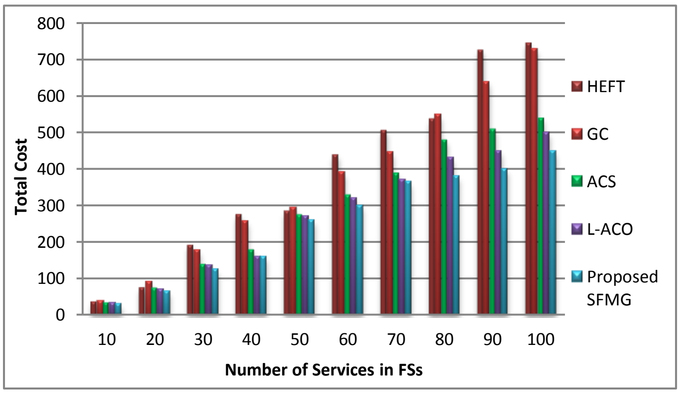

Fig. 4. Average consumed cost of HEFT, GC, ACS, L-ACO and proposed SFMG algorithms

The average completion time of the HEFT, GC, ACS, L-ACO and the proposed SFMG algorithms is presented in Fig. 5. The proposed SFMG algorithm has a less average consumed cost than HEFT, GC, ACS, L-ACO algorithms.

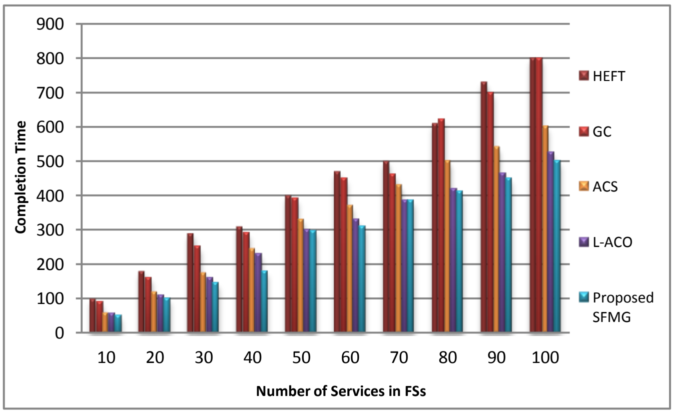

Fig. 5. Average completion time of HEFT, GC, ACS, L-ACO and Pproposed SFMG algorithms

*Schedule Length Ratio (SLR)*: it is a key measurement of a scheduling algorithm based on makespan. The schedule length ratio (SLR) is defined by Eq. (7) as in [1].

$$SLR = \frac{Makespan}{\sum_{v_i \in CP_{min}} min_{mt \in M}\{W_{i,t}\}} \qquad (7)$$

The divisor is the summation of services minimum computation on the *FS*. Low value of SLR means that the better performance of mapping algorithm. The SLR of the HEFT, GC, ACS, L-ACO and the proposed SFMG algorithms is presented in Fig. 6. The proposed SFMG algorithm has less SLR than HEFT, GC, ACS, L-ACO algorithms that prove the efficiency of the proposed SFMG algorithm.

*Speedup*: The speedup value is defined as the ratio of the sequential execution time to the parallel execution time and can be computed by Eq. (8) as in [7].

$$Speedup = \frac{min_{mt \in M}\{\sum_{v_i \in V} W_{i,t}\}}{Makespan} \qquad (8)$$

Where numerator represents the sequential execution time computed by assigning all services to a single candidate and the parallel execution time here is represented by makespan. The speedup of HEFT, GC, ACS, L-ACO and the proposed SFMG algorithms is presented in Fig. 7. The proposed SFMG has greater value of speedup measure than HEFT, GC, ACS, L-ACO algorithms. These results proves that the proposed SFMG has better performance.

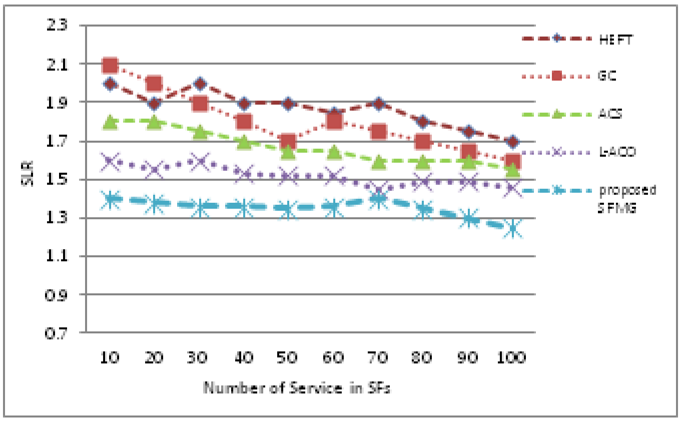

Fig. 6.   Average SLR of the Proposed GA, HEFT and GC Algorithms

The proposed SFMG algorithm outperforms HEFT, GC, ACS and L-ACO algorithms because it is able to search the solution space more efficiently based on models for minimizing total the total cost and minimizing the completion time. The ACS and L-ACO algorithms depend on ACO algorithm that uses group of ants in searching for the near-optimal management solution. So these two algorithms outperformed the HEFT and GC algorithms. The proposed SFMG algorithm deals with the consumed cost and completion time level as commensurable objective function that gather a two separated objective functions. Each one has its own measure that should be optimized. It consolidates the partial solution information under construction and the information of the best solution, therefore it can find more appropriate service flow allocation and achieve better performance.

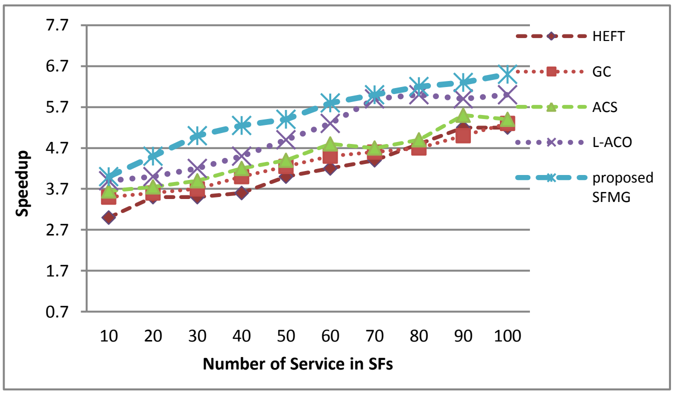

Fig. 7.   Average Speedup of the Proposed GA, HEFT and GC Algorithms

## VI.    CONCLUSIONS AND FUTURE WORK

This paper proposes a genetic algorithm for service flow with multi-objective constraints in heterogeneous computing that is called SFMG. SFMG handles service flow execution and management taking into account two objectives. They are the deadline and the cost. The SFMG aims to decrease the execution time objective to reunite the deadline constraint and reduce the consumed cost according to the user specified budget. A simulator based CloudSim is proposed to measure the efficiency of the SFMG algorithm. The proposed simulator generate a different service flows randomly that contains a various services from 10 to 100. It also creates a diverse candidates with a specified capability power processing and cost for each services.  The control parameters of SFMG is adapted experimentally.  For each measure, the related experiment is repeated more than ten times to get the average values. The produced results proves that the proposed SFMG algorithm overcomes HEFT, GC, ACS, L-ACO algorithms in terms of total cost, completion time, speedup and schedule length ratio measurements. The reliability and unbalanced service flow structure will be addressed in future work.

## REFERENCES

[1] Naqin Zhoua , Deyu Qi, Wei Feng, Xinyang Wang and Yang Shen, "Budget-Deadline Constrained Workflow Scheduling for Heterogeneous Resources", International Conference on Computational Science and Engineering (CSE), 2017.

[2] Monika Prajapati and Neetu Sharma, "Scheduling Service workflow in Hybrid Cloud", Proceedings of the 2nd International Conference on Communication and Electronics Systems (ICCES), 2017.

[3] Hui Liu, Dong Xu and HuaiKou Miao, "Ant Colony Optimization Based Service flow Scheduling with Various QoS Requirements in Cloud Computing", First International Symposium on Software and Network Engineering, IEEE, 2011

[4] Ashish Gupta and Ritu Garg, "Workflow Scheduling in Heterogeneous Computing Systems : A Survey", International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), 2017.

[5] Huangke Chen, Xiaomin Zhu, Dishan Qiu, Ling Liu and Zhihui Du, "Scheduling for Workflows with Security-Sensitive Intermediate Data by Selective Tasks Duplication in Clouds", IEEE Transactions on Parallel and Distributed Systems, 2016.

[6] Rodriguez M.A. and Buyya R., "A taxonomy and survey on scheduling algorithms for scientific workflows in IAAS cloud computing environments", Concurrency and Computation: Practice and Experience journal, vol. 29, 2017.

[7] Alkhanak EN , Lee SP , Rezaei R and Parizi RM, "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing", Journal of Systems and Software vol. 113, pp. 1–26, 2016.

[8] Henrique Yoshikazu Shishido, Júlio Cezar Estrella, Claudio Fabiano Motta Toledo and Marcio Silva Arantes, "Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds", J.Computers and Electrical Engineering, pp: 1–17, 2017.

[9] Toledo C , França P , Morabito R and Kimms A, "Multi-population genetic algorithm to solve the synchronized and integrated two-level lot sizing and scheduling problem", Journal of production resaerch, vol. 47, no. 11, pp: 97 – 119, 2009.

[10] Ye G, Rao R, Li M, "A multiobjective resource scheduling approach based on genetic algorithms in Grid environment", Fifth International Conference on Grid and Cooperative Workshops, Hunan, China, pp: 504–509, 2006.

[11] Wu F , Wu Q and Tan Y., "Workflow scheduling in cloud: a survey",. J Supercomput, vol. 71, no. 9, pp. 373-418, 2015.

[12] D. B. Tracy, J. S.Howard, and B. Noah, "Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, vol. 61, no. 6, pp: 810 - 837, 2001.

[13] J. Yu, R. Buyya, "Workflow Scheduling Algorithms for Grid Computing. Metaheuristics for Scheduling in Distributed Computing Environments", Metaheuristics for Scheduling in Distributed Computing Environments book, Springer, Berlin, Germany, 2008.

[14] Suraj Pandey1, LinlinWu1, Siddeswara Guru2, Rajkumar Buyya1, "A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”, Technical Report in Cloud Computing and Distributed Systems laboratory, The University of Melbourne Australia, October,2009

[15] Jianfang C , Junjie C and Qingshan Z , "An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm", Journal of Cybernetics and Information Technologies, vol. 14, no. 1, pp. 25–39, 2014.

[16] Zhangjun Wu, Xiao Liu, Zhiwei Ni, Dong Yuan, Yun Yang. " A Market- Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems", Journal of Supercomputing, Special issue on Advances in Network&Parallel Comptg, 2010.

[17] Quanwang Wu, Fuyuki Ishikawa, Qingsheng Zhu, Yunni Xia and Junhao Wen, "Deadline-constrained Cost Optimization Approaches for Workflow Scheduling in Clouds", IEEE Transactions on Parallel and Distributed Systems, 2017.

[18] Bin Xiang, Bibo Zhang, and Lin Zhang "Greedy-Ant: Ant Colony System-Inspired Workflow Scheduling for Heterogeneous Computing", IEEE Transactions on Software Engineering, vol. 5, pp. 11404 – 11412, 2017.

[19] Verma A and Kaushal S, "Budget constrained priority based genetic algorithm for workflow scheduling in cloud", In: Proceedings of the fifth international conference on advances in recent technologies in communication and computing, pp: 216–22, 2013.

[20] RHui Liu, Dong Xu and HuaiKou Miao, "Ant Colony Optimization Based Service flow Scheduling with Various QoS Requirements in Cloud Computing", First International Symposium on Software and Network Engineering, 2011.

[21] Ghalem, B., Fatima Zohra, T., and Wieme, Z, "Approaches to Improve the Resources Management in the Simulator CloudSim" International Conference on Information Computing and Applications, pp. 189–196, 2010.

[22] A. S. Wu et al., "An Incremental Genetic Algorithm Approach to Multiprocessor Scheduling", IEEE Transactions on Parallel and Distributed Systems, vol. 15, no.9, pp: 824-834, September 2004.

[23] H. Topcuoglu, S. Hariri, and M.Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing", IEEE transactions on parallel and distributed systems, vol. 13, no. 3, pp. 260–274, 2002.