

# DeepSched: A Deep Representation Of Scheduling Policies For Heterogeneous Distributed Systems

Mohamed Shawky | Remonda Talaat | Mahmoud Adas | Evram Youssef  
 SEC 2, B.N 16 | SEC 1, B.N 20 | SEC 2, B.N 21 | SEC 1, B.N 9

{mohamedshawky911, remondatalaat21, mido3ds, evramyoussef}@gmail.com

**Abstract—To be conducted, once the results are complete.**

## I. INTRODUCTION

THE continuous evolution of computing power and growth of widely-distributed applications and *Internet of Things* (IoT) have driven our needs for better resource allocation methods. The performance of any system critically depends on the algorithms used to schedule tasks on its resources. The general objective is to get the best possible performance with the reasonable resource utilization. Recently, the heterogeneity of the computational systems are increasing, due to the nature of the applications. A *heterogeneous system*[1] can be defined as a range of resources, different in underlying architecture. The usage of a heterogeneous system has proven to be very efficient in increasing the system performance and reducing the overall power consumption. However, this comes with the problem of complex task scheduling. The task scheduling problem for a heterogeneous computing system is more complex than that for a homogeneous system because of the different execution rates of processors and different communication rates among different processors. The parameters of such problem make it very suitable for various optimization techniques. Previous work has investigated the usage of techniques such as genetic algorithms[2] to solve the task scheduling problem.

In this paper, we investigate the ability of *neural networks* to learn a representation of static local task scheduling problem. The recent advances in *Deep Learning* research is used to build our scheduling system. *Heterogeneous Earliest Time First* (HEFT)[3] is used as a target, where the network learns to approximate its performance. Also, we include a genetic algorithm experiment for the same problem for comparison. We will discuss our approach to the problem and its limitations.

## II. RELATED WORK

Since task scheduling in heterogeneous systems is a wide problem, a lot of research has been conducted in this area.

Various techniques have been introduced for the problem. The scheduling algorithms for heterogeneous distributed systems are divided into two main categories based on their target metric. Application-Specific scheduling targets job execution speed (performance). System-Specific targets resource utilization.

*Heterogeneous Earliest Time First* (HEFT)[3] is one of the most well-established algorithms for task scheduling in heterogeneous systems. It's a local static scheduling algorithms, which deals with a *Direct Acyclic Graph* (DAG) of tasks displaying the dependencies between different processes. Each task has a running time for each different machine and a time for communicating the results to children tasks. A wide range of algorithms[4] follows HEFT further improving the scheduling performance.

Recent research has been conducted to use optimization algorithms and predictive models to schedule tasks on heterogeneous system, since these algorithms have proven to be very efficient in solving complex computational problems. *Genetic algorithms*[2] can be used to efficiently schedule tasks in different systems. Also, *Reinforcement Learning* (RL)[5] has been a strong candidate for the problem, as the scheduling problem can be formulated as an *RL* problem, solved by various *RL* techniques. Some papers also investigated the usage of *Artificial Neural Networks* (ANN)[6] in task scheduling in heterogeneous systems.

In this paper, we make use of recent advances in *Deep Learning* and build a deep representation of the scheduling problem using *Recurrent Neural Networks*(RNNs)[7] and other techniques.

## III. METHODOLOGY

### A. Formulation

First, let us specify the main target of this work. As the task scheduling in heterogeneous distributed system is a vast field with lots of settings and algorithms[4], we choose our scheduling setting to be local and static. The goal of this work is to explore the ability of neural networks to

approximate the performance of some well-established scheduling algorithms. In our setting, the algorithms are provided a list of tasks, each has its own execution time on different machines and a set of heterogeneous machines, on which the tasks are executed. We compare the results neural networks to some other heuristic and approximate methods.

Although, we choose local static scheduling, we assume that the neural approximation methods can work in other settings as well.

### B. Baseline

Two baseline methods are considered. We choose *Heterogeneous Earliest Finish Time* (HEFT)[3] to be our heuristic baseline, as it's one of the widely-used scheduling algorithms in heterogeneous systems. Also, genetic algorithms[2] is provided as an approximate baseline.

1) *Heterogeneous Earliest Time First (HETF)*: HEFT[3] is an old and well-established algorithm for scheduling in heterogeneous systems. It has been used as a baseline in research due to its near-optimal results. Basically, it chooses some process to run on some machine, if the former has the earliest finish time, with considering the special nature of the heterogeneous system. However, HEFT isn't practical for real-world systems, as no-one can know in advance the finish time of a process.

2) *Genetic Algorithms*: Genetic algorithms[2] have proven to be very efficient in solving some of the hardest optimization problem. Previous work on the use of genetic algorithms in task scheduling [2] is adapted to the heterogeneous setting by adding parameters for execution time of different tasks on different machines.

The targeted *fitness function* will be the average waiting time  $W_t$  for all tasks. The fitness function for a specific schedule  $S$  is given by:

$$Fitness(S) = \frac{\sum_{i=1}^N W_{ti}}{N} \quad (1)$$

### C. Neural Networks (Deep Learning)

The main novel idea of this work is the use of *neural networks* for the scheduling problem. Neural Networks are known to be very good at learning representations and function approximation. In this work, we propose an architecture that can be used to reach a good approximated solution for the scheduling problem.

Previous work tried to use neural networks to solve scheduling problem through *Hopfield Nets* and *Inhibitor Neurons*[6]. However, we try to make use of the recent advances in *Deep Learning* to redefine the scheduling problem and solve it.

As mentioned, The provided input is a list of tasks with its information and a set of machines. We want the network to learn an implicit representation of the given tasks and machines. Then, the network learn the mapping between the learned representation and the desired schedule. In other words, the target of the network is to take two input list of information of both tasks and machines and then output for each task, the machine to be executed on, the actual start time and the actual finish time.

1) *Architecture*: Our architecture consists of two encoders and one decoder, as shown in Fig.1.

The first encoder is a fully convolutional network of 1D convolutional layers, to which is machines specifications are passed. This encoder learns a representation for the machines to be able to use it later in producing the outputs. The second encoder is an *Recurrent Neural Network* (RNN)[7], to which the tasks are passed one by one ordered by the arrival time. This network learns a representation of the tasks preserving the arrival order. RNN is chosen due to its proven effectiveness in dealing with sequential data. The two representations are then stacked and passed to two modules. A classification module which states the machine on which a specific task is executed. A regression module that identifies the actual start time and the actual finish time. Mainly, the two modules consist of 1D convolutional layers.

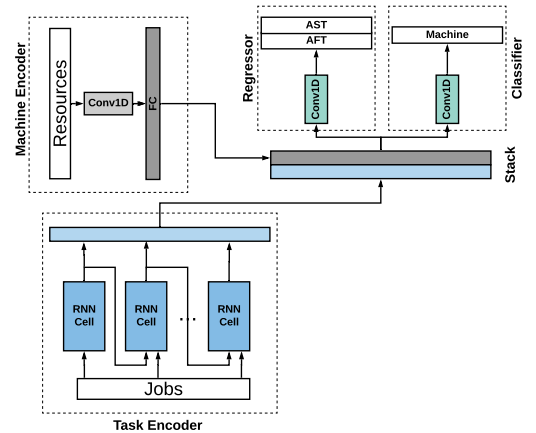


Fig. 1. The architecture of the proposed neural network for distributed heterogeneous systems scheduling

2) *Objective Function*: The cost function defined for this architecture is a sum of two criteria. Categorical cross entropy ( $CE$ ) on the predicted machine to run a specific job  $M_p$  and mean square error ( $MSE$ ) on the predicted actual start time  $AST_p$  and the predicted actual finish time  $AFT_p$ . So it can be summarized as follows:

$$Loss(L) = CE(M_p, M_t) + MSE(AT_p, AT_t) \quad (2)$$

where  $M_p$  is the predicted machine,  $M_t$  is the ground truth machine,  $AT_p$  is the predicted actual time range and  $AT_t$  is the ground truth actual time range.

3) *Downsides*: The main downside of the proposed solution is that we have to know the maximum number of machines and the maximum number of tasks to be able to define the network. These numbers are used to define the network and loss function dimensions.

#### IV. EXPERIMENTAL SETUP

#### V. DISCUSSION AND RESULTS

#### VI. CONCLUSION

#### REFERENCES

- [1] Ashish Venkat and Dean M Tullsen. Harnessing isa diversity: Design of a heterogeneous-isa chip multiprocessor. In *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, pages 121–132. IEEE, 2014.
- [2] Ranjeet Kumar, Dr.Rakesh, Er.Rajiv, Er Gill, and Ashwani Kaushik. Genetic algorithm approach to operating system process scheduling problem. *International Journal of Engineering Science and Technology*, 2, 09 2010.
- [3] H. Topcuoglu, S. Hariri, and Min-You Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, March 2002.
- [4] Dr. Mamta Padole and Ankit Shah. *Comparative Study of Scheduling Algorithms in Heterogeneous Distributed Computing Systems*, pages 111–122. 01 2018.
- [5] Alexandru Iulian Orhean, Florin Pop, and Ioan Raicu. New scheduling approach using reinforcement learning for heterogeneous distributed systems. *Journal of Parallel and Distributed Computing*, 117:292 – 302, 2018.
- [6] Daniel Chillet, Antoine Eiche, Sébastien Pillement, and Olivier Sentieys. Real-time scheduling on heterogeneous system-on-chip architectures using an optimised artificial neural network. *Journal of Systems Architecture - Embedded Systems Design*, 57:340–353, 04 2011.
- [7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.