Dynamic Task Scheduling with Unsupervised Self Organizing Map

ZHIMO ZHOU*, Universe of Guelph, Canada
UDAY SINGH†, Universe of Guelph, Canada
YAOWEN MEI‡, Universe of Guelph, Canada

Some Abstract I will add later

CCS Concepts: • **Social and professional topics** → **History of software**; • **Computer systems organization** → **Architectures**.

Additional Key Words and Phrases: Android Studio, Eclipse, Apache Maven, Gradle, Software Architecture

ACM Reference Format:
Zhimo Zhou, Uday Singh, and Yaowen Mei. 2021. Dynamic Task Scheduling with Unsupervised Self Organizing Map. 1, 1 (June 2021), 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Self-Organizing Map (SOM), also know as Kohonen map, is a topological preserving map that can map a higher dimensional space to a lower dimensional space. Along this process, information will be compressed; while, the key parameters in terms of "topological and metric relationships"[2] will be retained.

There are two steps involved in forming a self-organizing map from a raw input data-set[1], respectively to be 1) **competition** and 2) **cooperation**. When a set of data is feed into the system sequentially with random shuffle, for each input data point, **competition** will take place first and, based on a pre-defined cost function, one of the neurons on the output layer with the minimal cost will be selected as a winner; Following the competition, the **cooperation** will then take place. Based on a neighborhood function, the winner together with it's neighbor neurons will proceed the learning; while, the neurons outside of the winner's neighbor zone will gain no learning. The purpose of the cooperation step is to increase the like-hood that if a similar input pattern present again, the same group of neurons will become the winner. Iterate with this strategy on the input data-set over a suitable period, without supervising (providing error to the system), the output layer will simultaneously form a map that contains the similar topological structure as the input data.

*Both authors contributed equally to this research.
†Both authors contributed equally to this research.
‡Both authors contributed equally to this research.

Authors' addresses: Zhimo Zhou, zzhou17@uoguelph.ca, Universe of Guelph, 500 Stone Rd, E, Guelph, Ontario, Canada, N2L 3G1; Uday Singh, usingh03@uoguelph.ca, Universe of Guelph, 500 Stone Rd, E, Guelph, Ontario, Canada, N2L 3G1; Yaowen Mei, ywmei@uoguelph.ca, Universe of Guelph, 500 Stone Rd, E, Guelph, Ontario, Canada, N2L 3G1.

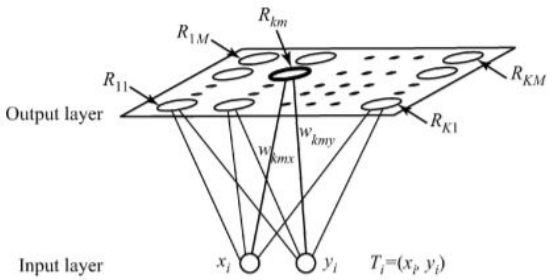

Fig. 2. SOM-based NN structure of the proposed approach, where input (x_i, y_i) denotes the ith target location, and output $(R_{11}, R_{12}, \ldots, R_{KM})$ denotes the positions of robots and the robot paths at a time instance.

Fig. 1. The illustration of the intermediate step to map K inputs (Robots) to M outputs (Targets)[3]. At a time instance, i, K neurons out of these $K * M$ possibilities will be selected as the winner. [I do not really understand this!!!, should we have K winner as we only have K inputs or M winner as each neuron should have same possibility to win???]

2 MODEL AND METHOD

- 1D model of K robots and M Targets
- 2D model of K robots and M Targets
- KD model of K robots and M targets

REFERENCES

[1] Colin Fyfe. 2007. *Hebbian learning and negative feedback networks.* Springer Science & Business Media.

[2] T. Kohonen. 1990. The self-organizing map. *Proc. IEEE* 78, 9 (1990), 1464–1480. https://doi.org/10.1109/5.58325

[3] Anmin Zhu and Simon X Yang. 2006. A neural network approach to dynamic task assignment of multirobots. *IEEE transactions on neural networks* 17, 5 (2006), 1278–1287.