Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. D. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

AN EVALUATION OF SPACE AND GRAPH-PARTITIONING METHODS FOR DISTRIBUTED ROAD NETWORK SIMULATIONS

Aravind Vasudevan[1], Quentin Bragard[2], Anthony Ventresque[2], Liam Murphy[2] and David Gregg[1]

[1] School of Computer Science and Statistics, Trinity College Dublin
[2] Lero@UCD, School of Computer Science and Informatics, University College Dublin

1 INTRODUCTION

We evaluate three different approaches to road network partitioning for simulation. The first approach consists of algorithms that rely on dividing the road physical network spatially into rectangular blocks of different sizes. Our second approach uses a graph representation of the road network, and uses meta-heuristic search algorithms to partition the graph. The final hybrid scheme builds clusters of nodes in the graph representation of the road network, based on spatial information. One of the most important contribution in this paper is the novel formulations of the well-known meta-heuristic approaches of simulated annealing and genetic algorithms to solve the road network partitioning problem. We also present a detailed experimental comparison of the three major approaches, using the road networks of several world cities (Table 1).

2 FORMALIZING THE OBJECTIVE FUNCTION

The road network of a city can be represented by a directed graph $G(V,E)$ where V denotes the vertex set and E denotes the edge set. We define three heuristics that are widely accepted in the community, to measure the effectiveness of a *partitioning scheme*.

One of the primary objectives of road network partitioning for distributed simulation is to minimize the execution time of the simulation. This is the case with any task-partitioning algorithm, where the slowest partition dictates the execution time of the application. To this end, we aim to minimise the weight of the **Heaviest Partition** (λ_G^{max}), by minimizing the following equation:

$$\lambda_G^{max} = max(w(P_i)), \forall P_i \in \{P_0, \ldots, P_{p-1}\} \tag{1}$$

The execution time of a distributed simulation also depends on the amount of **Inter-Partition Communication** (λ_G^{comm}). In order to reduce this inter-partition communication, we minimise the following metric,

$$\lambda_G^{comm} = \sum_{i \in \{v_0,\ldots,v_{|V|-1}\}} \sum_{j \in \{v_0,\ldots,v_{|V|-1}\}} w(e_{ij}) \tag{2}$$

The final metric that we examine is the **Evenness** (λ_G^{σ}) of the partitions. Consider a scenario in which the underlying parallel computer architecture consists of homogeneous execution units with a powerful communication backbone. In this scenario, a vertex of the road network graph takes equally long to simulate on any machine in the underlying parallel computer architecture. The **Evenness** metric plays a very important role here as it defines how varied each machine is loaded.

$$\lambda_G^{\mu} = \frac{\sum_{P_i \in \{P_0,\ldots,P_{p-1}\}} w(P_i)}{p} \qquad \lambda_G^{\sigma} = \frac{\sqrt{\sum_{P_i \in \{P_0,\ldots,P_{p-1}\}} (w(P_i) - \lambda_G^{\mu})^2}}{p} \tag{3}$$

3 META HEURISTICS BASED GRAPH PARTITIONING ALGORITHMS

Table 2 gives a list of acronyms, for the three major approaches and its variants, that we compared based on the cities listed in Table 1. We employ genetic algorithm with two variants which are widely explored in the literature. We also employ four variants of the simulated annealing method. All these variants differ in the definition of their **move** function. Our baseline is the conventionally accepted notion of moving through the search space using random neighbours. To address the issue of expensive inter-partition communication, we introduce the concept of **Edge-Labelling**. We label edges as **heavy** or **light** by virtue of their edge weights as compared to the *rest of the edges' weights*. We introduce further variants by restricting the scope of edge labelling thereby giving rise to : **SA-GE**, **SA-LE-R** and **SA-LE-G** from Table 2

Cities	$\|V\|$	$\|E\|$
Barcelona	13,476	25,658
Kyoto	42,456	93,722
San Francisco	15,436	35,092
Cologne	56,548	115,483
Lyon	8,174	15,586
Miami	8,141	21,856

Table 1: The task graph setup

	Algorithm	Acronym
Space Partitioning	TwoTree	**TT**
	Smart QuadTree	**SQT**
Meta-heuristics	Standard Simulated Annealing	**SA**
	Global Edge Labelling	**SA-GE**
	Local Edge Labelling — Random	**SA-LE-R**
	Local Edge Labelling — Guided	**SA-LE-G**
	Genetic Algorithm — Mutation	**GA-M**
	Genetic Algorithm — Crossover	**GA-C**
Hybrid Scheme	SParTSim	**SPS**

Table 2: Acronyms for algorithms

4 EXPERIMENTS AND RESULTS

Due to the enormity of the search space, the meta-heuristic algorithms perform very poorly with a random initial solution. Hence, we use the result from **SQT** as the starting solution. Table 1 presents an overview of the cities that are used to compare the different partitioning strategies. When comparing the methods under the *Evenness* metric (eq. 3) the space partitioning methods do not perform well while some of the meta-heuristic methods do well. **SPS** outperforms **SQT**, **TT** and the **GA** methods, except in the case of Kyoto and Cologne. All the variations of SA perform well producing results that are $\sim$99% better than **SQT** while the two variations of GA give stunted improvements($\sim$7%) in comparison. **TT** performs the worst of the two space partitioning methods being an order worse than **SQT**, for small number of partitions.

Under the *Inter-Partition Communication* metric, random search heuristics do not perform well because of the enormity of the search space. However, we made an interesting observation for Miami and San Francisco where **SA-LE-G** produces a partitioning scheme which is $\sim$87% better than its parent curve(**SQT**). The hybrid method **SPS** suffers from its `Trade` method and is on average 22% worse than the space partitioning methods.

Finally, under the *Heaviest Partition* metric, **TT** performs the worst as it does not perform any form of vertex trading. This is a direct result of high dependence on spatial information. The *Local Edge Labelling* variants of SA : **SA-LE-G** and **SA-LE-R** perform consistently better than all the other competitors. Interestingly, the roles for these two methods have been reversed. Under eq. 3, **SA-LE-R** performs better than **SA-LE-G** in 82% of the cases while under eq. 1, **SA-LE-G** performs better than **SA-LE-R** in 98% of the cases.

5 CONCLUSION

We find that the Space Partitioning methods perform very well for minimizing *Inter-Partition Communication* while one of the Meta-heuristics **SA-LE-G** produces astounding results for select cities. However, the simplicity of the Space Partitioning algorithms and its inability to trade vertices of the road network graph prevent it from neither load balancing (*Evenness*) well nor effectively reducing the size of the *Heaviest Partition*. The Meta-heuristics on the other hand perform very well for balancing the load on different partitions.