Chapter 5

Experiments

To experimentally verify the effectiveness of emergent optimization, a grid is simulated with 64 nodes arranged in a $8 \times 8$ square. The 12 source nodes are distributed around the center of the grid along the periphery of a smaller square as shown in Figure 5.1. 10000 queries arrive uniformly on the grid nodes at a rate of one query every unit of time. Queries seek data or a subset of the data available at the source nodes. Each query remains in the grid for a certain amount of time and is then terminated.

It should be noted here that only 64 grid nodes were considered as, the primary objective of the experiments performed was to evaluate and compare the effectiveness of emergent query plan optimization and not test the scalability of emergent optimization. Further more, computing the globally optimal query plan for each query and de-query is computationally intensive. It should also be noted that the 12 sensor nodes result in 4095 distinct query possibilities, which are used
(with equal probability) to generate the set of queries incident on the grid.

![Grid Layout Diagram](image)

**Figure 5.1: Experimental Grid Layout**

In the experiments we first evaluate the effectiveness of emergent optimization using the DO strategy in reducing network usage when compared with globally optimal query plans generated by taking snapshots of the grid with every new query arrival and query termination. We then confirm the ability of the LO strategy in distributing load evenly. We also demonstrate the ability of the grid nodes to identify the correct strategy for a given global optimization objective.

However, in reality, the optimization objective is not set by any grid administrator but will depend on the queries incident on the grid. If there are a relatively small number of queries, the grid would need to optimize on network usage and ignore load distribution, but when the average number queries increase beyond a threshold, the grid would need first identify the change in the query pattern and select the correct strategy which would enable it to balance the load on the grid.
The payoff function used in the experiments and its rationale is described in the next section.

5.1 Payoff Function

For the experiments we use payoff function that has two payoff components, one of which is based on network usage and the other on the resulting load on data sources. We choose a payoff function that results in, (1) higher payoffs for a node if its source selection leads to reduced network usage and, (2) reduction in payoffs, with the reduction proportional to the load on the data sources selected. The details of the payoff function are given below.

A query $q_i$ brings with it a certain amount of virtual currency or income, $I(q_i)$ which is equal to the network usage if the query were to be answered directly from the source nodes using a set of links $L(q_i)$.

$$I(q_i) = \sum_{l_i \in L(q_i)} u(l_i)$$  \hspace{1cm} (5.1)

such that all $\text{source}(l_i)$ are primary sources.

As mentioned earlier, the payoff function $\rho$ is split into two parts, (a). a network usage part, and (b). a load distribution part. If $L(q_i)$ is the set of links using which $q_i$ is answered, the network usage part $\rho_U$ for answering a query $q_i$ on node $x$ at time $t$ is the ratio of unused virtual currency to the total virtual currency and
is given as,

$$\rho_U = \frac{1}{I(q_i)} \cdot [I(q_i) - \sum_{l_i \in L(q_i)} u(l_i)] \quad (5.2)$$

while the load distribution part \( \rho_W \), is measured as the average load ratio over all the links required to answer a query and is given by,

$$\rho_W = \frac{1}{|L(q_i)|} \sum_{l_i \in L(q_i)} \frac{w_{\text{source}(l_i), \mathcal{L}}}{\text{MAXLOAD}(\text{source}(l_i))} \quad (5.3)$$

where \( \text{MAXLOAD}(\text{source}(l_i)) \) is the maximum number of links node \( \text{source}(l_i) \) can handle.

The overall payoff \( \rho \) for the query is evaluated as,

$$\rho = \alpha \cdot \rho_U - (1 - \alpha) \cdot \rho_W \quad (5.4)$$

where, \( \alpha \in [0, 1] \) is a configurable parameter that determines the intrinsic importance to be given by the grid to network usage or load distribution.

Since \( \rho_U \) and \( \rho_W \) are both ratios, they can be added together in a single equation. However, they may not have the same characteristics and the impact of \( \alpha \) on \( \rho_U \) and \( \rho_W \) need not be the same. In our grid scenario, a value of 0.2 for \( \alpha \) was empirically seen to provide equal importance to network usage and load distribution.

In prior work, Stegmaier et al. [80] and Kuntschke et al. [46], the value of \( \alpha \) needs to be set by the grid administrator to reflect the preference for optimizing either on network usage or load distribution. However, it should be noted here
that in StreamGlobe, $\alpha$ cannot be a static parameter fixed to optimize under all possible conditions in the grid. Consider a small set of queries incident on the grid and therefore do not load the nodes in the grid. Under such circumstances, it does not make sense to consider load distribution. Similarly, if there are a lot of queries and the nodes in the grid are heavily loaded, then it does not make sense to consider network usage during optimization. Hence $\alpha$ needs to be adjusted according to the varying query patterns incident on the grid. This would entail that the grid be monitored continuously and $\alpha$ adjusted to meet the varying query patterns. Finally, it would also be difficult for the administrator to predict the exact value of $\alpha$ which would result in the required global optimization objectives being met.

In contrast, in our work, the emergent nature of the optimization ensures that the grid itself adjusts to the varying query patterns incident on the grid and the administrator needs to set $\alpha$ just once during the initial setup based on the grid characteristics.

### 5.2 Results

To evaluate the performance of emergent optimization using the DO strategy, we compare the instantaneous network usage or the network usage at every instance of time with the network usage resulting from the globally optimal query plan as described earlier. We also evaluate the load distribution abilities of the emergent strategy.
Since we are interested in measuring the performance of the various strategies and the ability of the grid to correctly select the correct strategy for a given optimization objective, we switch $\alpha$ between extremes, 1 (payoff only for optimizing network usage) and 0 (payoff only for optimizing load distribution) to indicate the optimization objective.

The grid nodes are provided with the three single strategies: DO, RO and LO. At the start of the experiment, nodes select a strategy with equal probability.

5.2.1 Instantaneous Network Usage

Figure 5.2 plots network usage across time, for the globally optimal plan in comparison with emergent optimization when the objective is to optimize solely on network usage ($\alpha = 1$), and when the objective is to optimize solely on load distribution ($\alpha = 0$).

In these experiments, a generation comprised of 250 time steps. We can see that emergent optimization comes very close to the globally optimal network usage when $\alpha = 1$. The network usage starts off with a high value given the random nature in which strategies are chosen. However, the demographics stabilize very quickly and the network usage drops very close to the optimal usage. When $\alpha = 0$ however, network usage is continually high, since the emphasis is on load distribution. This confirms the effectiveness of the emergent strategy using DO strategy for reducing network usage.
5.2.2 Instantaneous Load Distribution

Figure 5.3 compares the instantaneous load distribution characteristics when the global optimization objective is to optimize on network usage ($\alpha = 1$) and when the objective is to optimize on load distribution ($\alpha = 0$). The standard distribution of load is around 4 when the objective is to optimize network usage and around 2 when the objective is optimize load distribution, thereby indicating that the LO
strategy is better at balancing load compared to the DO strategy.

Figure 5.4 and Figure 5.5 plots demographic change of each of the strategies (DO, RO and LO) change across generations when $\alpha = 1$ and when $\alpha = 0$ respectively.

When $\alpha = 1$, we observe that all the nodes converge to the DO strategy while the LO and RO strategy becomes non-existent in the grid. Similarly, when $\alpha = 0$, we observe that all the nodes converge to the LO strategy. This demonstrates the ability of emergent optimization to select the correct strategy for a given optimization objective. It should be noted here that when $\alpha$ is 0 or 1, nodes do not select the RO strategy. This essentially shows that strategies which do not perform well for a given objective will be rooted out of the system.

5.3 Generational Dynamics

As seen in Figure 5.5 and Figure 5.4, the convergence towards a particular strategy is quick and leads to one of either the distance ordering or load ordering strategies surviving. Since the strategy selection is based on payoffs for a particular strategy, if the number of grid nodes adopting a particular strategy drops to zero, the grid will never consider that strategy because the payoff for such a strategy is always zero. This might lead to the grid being unable to adjust the varying conditions of the grid and query characteristics.

To mitigate this issue, grid nodes perform a strategy reset after some specified time “random interval” and select strategies at random to evaluate the possibility
of an alternate strategy being better than the one in use currently.

As mentioned earlier, the optimization objective in the grid is determined by the queries incident on the grid and the grid nodes need to identify and optimize with changing query patterns incident on the grid. To simulate fluctuating query patterns, we alter the average load on the grid by varying the average duration for which a query remains in the grid while keeping the arrival rate of queries constant. In the experiments performed the grid is subjected to three different query sets as shown in Figure 5.6 with varying average query loads. All the query sets have an increase in query load between time interval 4000 and 6000 to simulate changing query patterns.

![Figure 5.6: Query Sets Used for Experimentation](image)

The distribution of strategies in the grid nodes for the three different query sets is shown in Figure 5.7, 5.9 and 5.11. We use bezier curves to smooth out the disruptions caused by the strategy resets which occur at instances as indicated by the vertical lines in the graphs. The value of $\alpha$ was set to 0.2 for all the experiments.

Query set 1 (Figure 5.6) has the least average load and the fluctuation in load is
not significant enough for the grid nodes to change their strategy and hence and as seen in Figure 5.7 all the nodes select the DO strategy throughout the experiment and optimize on network usage.

When the grid is subjected to query set 2, the nodes select the DO strategy initially when the average load is around 200 and switch to the LO strategy to balance the load when the average load increases to 500. Once the average load decreases to 200, the grid nodes switch back to the DO strategy and continue to optimize network usage (Figure 5.9). This clearly shows that emergent optimization is able to identify and adapt to the changing grid conditions without any manual intervention.

In query set 3, the initial grid load is around 500 and hence the nodes adopt the LO strategy to balance the load. The nodes continue to use the LO strategy for the entire duration of the experiment Figure 5.11 as the load on the grid is remains high. Figures 5.8, Figure 5.10 and Figure 5.12 show the actual distribution of strategies in the grid for query sets 1, 2 and 3 respectively.

This set of experiments indicate that the principle of emergent optimization is able to select the best strategy among the set of strategies it is provided with and is also able to address the varying query patterns incident on the grid by switching from one strategy to another without any intervention.
Figure 5.7: Smoothed Strategy Distribution for Query Set 1

Figure 5.8: Exact Strategy Distribution for Query Set 1

Figure 5.9: Smoothed Strategy Distribution for Query Set 2

Figure 5.10: Exact Strategy Distribution for Query Set 2
Figure 5.11: Smoothed Strategy Distribution for Query Set 3

Figure 5.12: Exact Strategy Distribution for Query Set 3