Efficient Information Retrieval Using Mobile Agents

Irene Sygkouna
National Technical University of Athens
9, Heroon Polytechneiou Str., 15773, Zografou, Athens, Greece
isygk@telecom.ntua.gr

Miltiades Anagnostou
National Technical University of Athens
9, Heroon Polytechneiou Str., 15773, Zografou, Athens, Greece
miltos@central.ntua.gr

ABSTRACT

We are concerned with the use of Mobile Agents for information retrieval. A multi-agent system is considered; a number of agents are involved in a collective effort to retrieve distributed data from network nodes. We are seeking the routes of the agents that minimize the total completion time. The system design goal of producing an "optimal" answer with minimal use of communication and processing resources is the driving reason for testing three heuristic algorithms.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems – *Routing and layout;* G.2.2 [**Discrete Mathematics**]: Graph Theory – *Graph algorithms, Network problems;* H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *Information networks, Performance evaluation (efficiency and effectiveness)*.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Information Retrieval, Mobile Agents.

1. INTRODUCTION

Mobile Agents have come to the foreground as a promising approach for building distributed applications. Code mobility has proved to be even more profitable in information-intensive applications. A commercial transaction in E-Commerce usually requires real-time access to remote resources such as product catalogues and retrieval of large quantities of data for the sake of repository update of the menu of services managed by a client-mediator, which is responsible to serve timely customer requests [4]. In order to accelerate the retrieval of distributed information, we consider a multi-agent system, where the client initiates simultaneously multiple homogeneous mobile agents, each of which visits a subset of the providers that host the needed information. The agents then return to the client with the information requested to complete the task.

To address the problem of finding the less time consuming solution for the accomplishment of the information retrieval requirements, the definition of an optimisation problem is required. The objective is to find the routes traversed by a specific number of agents so that the total completion time is minimized, given that data from a node is retrieved by exactly one agent and besides at the last occurrence of the node in the route. The problem is defined on an undirected graph G(V,E), where $V = \{0,1,\dots,N\}$ is the set of nodes and E the set of edges of the graph. Let us also denote by $V' \subseteq V \setminus \{0\}$ the set of nodes that host the requested by the client data. Without loss of generality we assume that node 0 represents the client that launches the agents. We also assume that L represents the initial size of each mobile agent, l the size of data collected from a node, τ_{ij} the transmission rate of edge (ι, j) and K the number of agents launched. Moreover, we denote by z_{ik} the binary decision variable that becomes equal to 1 if data of node i is retrieved by agent k, whereas x_{ijk} determines whether agent k travels directly from node i to j or not.

The objective of this problem is to minimize the time of the agent, which completes its data retrieval task the latest. Supposing that L_{ik} corresponds to the accumulated size of the agent k after having visited node i, $\pi(i)$ shows the predecessor of node i in a route and considering only transmission delays, the objective function is formulated as follows:

Minimize

$$\max_k \sum_{i=0}^{N} \sum_{j=0}^{N} \frac{1}{\tau_{ij}} \cdot L_{ik} \cdot x_{ijk}, \qquad k = 1,\dots,K, \qquad (1)$$

$$L_{ik} = \begin{cases} L, & \text{if } i = 0 \\ L_{\pi(i)k} \cdot x_{\pi(i)ik} + l \cdot z_{ik}, & \text{if } i \in V' \\ L_{\pi(i)k} \cdot x_{\pi(i)ik}, & \text{otherwise} \end{cases} \qquad (2)$$

The problem in question shares some similarities with the Vehicle Routing Problem (VRP) [1]. However, a considerable difference derives from the fact that when time is considered the cost of traversing an edge depends on the accumulated volume of data retrieved so far and is therefore not fixed, whereas in the VRP the cost of traversing an edge corresponds usually to distance and is therefore considered fixed. Moreover, our problem is solved for a non-complete graph. The triangle inequality does not hold and an agent is likely to visit a node even without retrieving its data.

2. SOLUTION ALGORITHMS

As a generalization of the Travelling Salesman Problem, the problem under consideration is NP-hard. Thus, we applied three heuristic methods. *Savings* [2] and *Insertion* [3], which come from the VRP research area, belong to the category of *Constructive heuristics* since they gradually build a feasible solution while keeping an eye on solution cost. Initiating from the maximum possible number of routes, *Savings* performs repeated mergings until the number of routes becomes equal to the number of agents. In each iteration the most advantageous merging is kept. Merging route *A* with route *B* implies connecting the last node of route *A* from which data is retrieved to the first node of route *B* from which data is retrieved, through the shortest path. The basic idea of the *Insertion* heuristic is the gradual assignment of nodes to routes. Initiating from the minimum number of routes (equal to the number of agents), in each iteration the node that causes the most advantageous insertion is selected, trying sequentially all the possible places of each route. In each case, the most advantageous operation is considered as the one that produces the shortest route (in time).

The *Serial Reverse-Constructed Route* (SR-CR) heuristic is specially designed for this problem and attempts to find a solution by constructing the routes in a serial way while the construction of each route grows from back to forth, exploiting the following view: the more the agent size, the faster the links that the agent traverses should be. For this reason, starting from the back end from a route (node 0), it selects the next data node it will serve as the one that is reached through the shortest route (in time). Note that in all the above heuristics, the shortest path in terms of time from one node to another in case that no data retrieval is performed in the meanwhile, is computed with the assistance of *Dijkstra Shortest Path* algorithm, assuming that the weight of each edge $(i,j) \in E$ equals to $1/\tau_{ij}$.

3. EXPERIMENTAL RESULTS

In order to experience the behaviour of the algorithms we implemented a simulation in Java. The computational tests were performed on a set of network graphs that were created using the Waxman model [5]. We note that all the graphs used for the testing have the same density ($|E| \cong 1/3|V|^2$) and the transmission rate of each link is in the range (1,10) *Mbps*. Studying first the case where all nodes host data, we initiate out tests with a graph consisting of 60 nodes and we vary the number of agents from 6 to 42. The agent size equals to 20Kbytes and the size of data retrieved from each node equals to 5 *Kbytes*. Following, we increase the number of nodes to 80, and accordingly we vary the number of agents. We observe the following (Figure 1):

When the number of agents is high relative to the number of nodes (approximately $K/N > 0,4$), the Savings tends to produce better solutions than the Insertion or the SR-CR, whereas the SR-CR produces also interesting solutions. But as the number of agents decreases, the solutions produced by Savings deteriorate significantly rendering prohibitive its application to the problem. Thus, a drawback of the Savings is that it tends to produce good routes at the beginning but less interesting towards the end, where the number of routes has decreased due to the repeated mergings performed. For the latter case, namely when the number of agents employed is low, SR-CR algorithm seems to produce the best solutions whereas Insertion comes second.

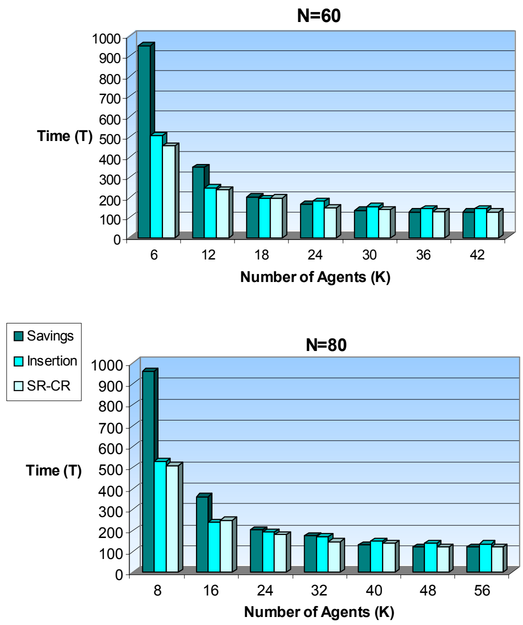

Figure 1: *Time* versus *Number of Agents*, for N=60 and N=80

Considering as performance metric the execution time, we observed that all the algorithms demonstrate the same order of growth with the number of nodes, whereas Savings' and Insertion's computation time decreases and SR-CR's increases with the number of agents employed. It is also noticeable that the computational requirements of Insertion become prohibitive for a relatively high number of nodes and a low number of agents.

4. REFERENCES

[1] Bertsekas, D. P., "Network optimization: Continuous and discrete models", Athena Scientific, 1998.

[2] Clarke, G., and Wright, J. W., "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, Vol. 12, pp. 568-581, 1964.

[3] Mole R. H., and Jameson, S. R., "A sequential route-building algorithm employing a generalized savings criterion", *Operational Research Quarterly*, Vol. 27, pp. 503-511, 1976.

[4] Valera, F., Kaltabani, A., Vazquez, E., Bellido, L., Sygkouna, I., Strimpakou, M., and Anagnostou, M., "Experiences with middleware and mobile agents in an e-commerce European project", *International Journal of Internet and Enterprise Management*, Vol. 2, No. 2, pp. 163-177, 2004.

[5] Waxman, B. M., "Routing of multipoint connections", *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 9, pp. 1617-1622, December 1988.