A Meta-heuristic based Multi-Agent Approach for Last Mile Delivery Problem

Maram Hasan and Rajdeep Niyogi

Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Uttarakhand, India

Keywords: Horizontal Collaboration, Multi-Agent System, Logistics, Last Mile Delivery Problem, Egalitarianism.

Abstract: e-Commerce has become a primary part of any country’s economy, and seeking maximum efficiency and level of service is an essential concern for any corporation in order to stay in business. Logistics has a significant impact on the efficiency of online transactions, especially with the increasingly competitive domain with minimal profit margin-left. Thus, the collaboration between many logistics service providers (LSPs) at different levels has become a desirable approach to reduce the overall costs and increase the utilization level of their resources. In this work, we propose a domain-independent multi-agent framework that allows different LSPs to plan their operations jointly. The system considers the individual satisfaction of LSPs and their profits in an egalitarian manner while trying to achieve an overall benefit. We use different search strategies for every agent as the underlying solving method, and investigate to what level taking the personal interest of participants will affect the overall shared/common goal.

1 INTRODUCTION

e-Commerce has grown remarkably over a short period of time and become an essential part of the industry (Chatterjee et al., 2016). In order to guarantee the success of an enterprise, companies need to preserve their reputation and trust of their customers and increase the quality of service they provide. In this matter logistics, or what is called the Last Mile Delivery (LMD), plays the most significant role as it refers to the last step in the supply chain (Holdorf and Haasis, 2014), in which the items are carried from the fulfillment centers to the customers within the specified delivery period. It forms the most important stage in the purchase process as it adds up nearly to 30% of the total cost. Thus, making this stage most effective is a critical issue in e-commerce transactions (Chinh et al., 2016).

Recently, all aspects of what is expected from logistics have changed, the competition became high, and profit margins declined to a deficient level. This urged many companies and carriers to collaborate on different levels and form partnerships in order to manage the transportation of their goods effectively (Ouhader and El Kyal, 2017); horizontal collaboration is a form of collaboration between parties that belong to the same level of the supply chain (all carriers or all suppliers). Collaborative routing helps to reduce the overall transport costs and to increase the customer satisfaction and utilization level of the logistics and human resources. Hence, it provides better performance from an economic and environmental perspective. For example, the collaboration between different LSPs can save them millions out of the total travel costs by sharing the logistics resources and thereby solving the empty load problem caused by the large number of vehicles that travel long distances underutilized (Chinh et al., 2016). This, in turn, reduces congestion levels on the roads, the possibility of accidents and harmful emissions, respectively (Ouhader and El Kyal, 2017). Figure 1a shows an example of independent routing of two LSPs in LMD where every LSP has one deposit. Meanwhile, Figure 1b shows a collaborative routing that utilizes the resources of each LSP to deliver the packages of all collaborators. It leads to overall shorter trips and lower costs.

As the transportation field is dynamic, complex and consists of many features and constraints, it makes the agent-based paradigm a quite suitable and exciting research method to solve famous problems in this field (Martin et al., 2016). Multi-agent frameworks are powerful in modeling complex problems of distributed nature and can efficiently represent the complicated relationship among the entities in an independent and autonomous way (Czarnowski et al.,...
A multi-agent system consists of a set of entities (agents) that interact with each other in some environment through predefined means of communication in order to accomplish some given goal(s) (Czarnowski et al., 2013). Agent-based paradigm increases the computational power and speeds up the performance as it enables parallel task executions; cooperation among the agents allows efficiency, reliability, and robustness. Thus, it is ideal for solving different complex optimization problems such as our last mile delivery problem.

Since most researches on horizontal collaboration focused solely on utilitarianism, which mainly gives importance to overall benefit (Cruijssen et al., 2007), (Lahyani et al., 2015), and since real-life collaborators mostly care about their profits and interests rather than the overall goal, our main contribution is to consider the satisfaction of individual participants in an equal manner and to investigate how this affects the global goal of reducing costs. For this purpose, we propose a domain-independent multi-agent framework in which each LSP agent represents a collaborator and manages a different search strategy to solve the LMD problem from its perspective while cooperating with other agents to combine and improve solutions of all agents. This paper is organized as follows; Section II presents a literature review of collaborative and multi-agent approaches to the problem of last mile delivery. Section III presents problem formation while Section IV explains the proposed framework. Section V discusses the results obtained; Section VI states the conclusion reached.

2 LITERATURE REVIEW

LMD can be considered as a Vehicle Routing Problem (VRP), which is a significant NP-hard that was first introduced in 1959 (Lahyani et al., 2015). It is concerned with finding the most cost-effective set of routes starting from a single distribution center to different customers spreading on a large geographic area while satisfying constraints (Kumar and Panneerselvam, 2012). Different versions of the problem were proposed in the literature under various complexities and constraints; (Kumar and Panneerselvam, 2012) and (Lahyani et al., 2015) present helpful surveys. Various methods have been proposed by researchers to solve the problem, starting with exact methods (Mingozzi et al., 2013), and heuristics and meta-heuristics that find near-optimal solutions (Clarke and Wright, 1964) to hybrid meta-heuristics such as hybridized genetic algorithm (Cattaruzza et al., 2014).

2.1 Horizontal Collaboration

With the increasing market demands and high customer expectations, horizontal collaborative routing has been studied to provide efficient routing with lower total operational costs (Chinh et al., 2016). Different methods were presented through time starting with centralized approaches where a centralized entity solves the optimization problem on behalf of collaborating LSPs with different levels of coordination (Chinh et al., 2016). However, in most cases, there is less willing of different participants to share their full customers’ details with a central authority; thus decentralized approach was presented in the literature to provide more compatibility with these privacy concerns such as request exchange and auction-based methods. The work (Wang and Kopfer, 2015) proposed two distributed request exchange mechanisms to solve the dynamic routing problems to provided stable performance. While in (Wang and Kopfer, 2014), a route-based exchange mechanism between collaborating LSPs that do their own planning independently and then send the planned routs to a groupage system (GS) which regenerates routes to reduce the overall costs. Meanwhile, auction-based trading is presented in (Dai and Chen, 2011) to al-
low the participating LSPs to negotiate and express their preference and the profit margin in a multi-round fashion. Another auction-based clustering method was proposed in (Schwind et al., 2009), where a request is exported based on its cost-reduction probability using a convex hull method with a distance-based cost estimator. Meanwhile, our proposed method is decentralized and allows LSPs to jointly plan their pre-defined outsourcing requests and continuously exchange existing solutions between computing agents to improve the final solution.

As the works mentioned above considered achieving a global goal of reducing overall cost, other work in (Zhou et al., 2013) focuses on load balancing in terms of the tour distance among the moving vehicle as a global goal; wherein (Schwarze and Voß, 2013), they examined effective resource utilization using a minmax model to minimize the maximal vehicle trip length. In our work, we consider the satisfaction of individual participants in an equal manner while achieving total cost reduction.

2.2 Agent-based Approaches

Agent-based systems have contributed to collaborative routing solving because of its autonomous, distributed characteristics and its native support for a dynamic environment (Czarnowski et al., 2013). Several domain-specific multi-agent architectures were proposed in the literature, such as in (Baykasoğlu and Kaplanoğlu, 2015), where the architecture of three types of agents (order, truck, dispatcher) was proposed to obtain effective resource utilization under dynamic environment’s conditions through negotiation. Also, in (Kalina et al., 2015), they provided a multi-agent system consists of a set of customer agents, route agents, and a central planner. The planner used a local heuristic search to solve the routing problem with time windows. In contraction, our work will not be dedicated only to LMD domain as our agents will be optimizing agents that able to work on any optimization problem. An example of earlier population-based, cooperative multi-agent system was proposed in (Barbucha, 2014); where a multi-phase execution is carried out and different search meta-heuristics are executed at each phase simultaneously while exchanging information about the agents’ states and performance at the end of each phase. In (Martin et al., 2016), they proposed a general multi-agent approach where agents periodically exchange parts of their solutions that have potentials to be good edges to guide the search of other agents in a valuable direction. Also, the work (Chatterjee et al., 2016) included public transportation system in their multi-agent frame-

work to plan small-size orders delivery in aim to reduce costs and CO2 emissions.

The work in (Barbucha, 2012) used different synchronization modes in their proposed multi-agent system, and used a learning mechanism to adapt the behavior of the agents to new states of the environment while solving the optimization problem. Meanwhile, the work (Souza et al., 2012) used particle swarm optimization (PSO), where each particle is considered as autonomous agents and represented as a solution in the search space. Otherwise, our proposed multi-agent framework consists of a group of optimization agents exploring different areas of the search space simultaneously, producing a higher level of diversity and mature solutions.

Although collaborative routing has been studied in past years, there are fewer studies in the literature as far as we know that use a multi-agent paradigm to solve a multi-deposit VRP version that considers the personal interests of participants in an equal manner along with a global goal. The main contributions are as follows; we implement a multi-agent framework that supports the collaboration between different LSPs to serve their customers jointly. Our framework is domain-independent, which can be used to solve any optimization problem in contrast to (Baykasoğlu and Kaplanoğlu, 2015), (Kalina et al., 2015). Agents will regularly exchange the best solutions among themselves to combine and improve the solutions.

3 PROBLEM DEFINITION

Last Mile Delivery Problem can be presented as a vehicle routing problem. It aims to create a set of paths to satisfy the delivery of \( n \) packages, from distribution center \( s \) to specified delivery locations satisfying some constraints ex. capacity and time, while minimizing a cost function. Each Logistic Service Provider LSP has a set of distribution centers \( DCs \) which contains numerous different-size packages to be distributed to many geographically scattered customers within a city. Each package has a weight demand \( Dem_i \). A distribution center has a set of vehicles \( V \) used to complete the delivery tasks, each of these vehicles \( v \in V \) has a capacity \( Q_v \) that specify the maximum load a vehicle can carry at once.

Here we present the notations of the problem; given a undirected graph \( G = \langle V, E \rangle \), the set of vertices \( V \) represents the customers, and the set of edges \( E \) determines the existence of a direct road between the adjacent customers; \( D \): set of distribution centers \( D = \{1, 2, 3, \ldots, m\} \); \( C \): set of customers \( C = \{1, 2, 3, \ldots, n\} \).
\[ \{1,2,3,\ldots,n\}; K \text{ is the total count of vehicles existing in all of the distribution center.} \]

\[ V_{DC} : \text{sets of vehicles presented in the distribution center } DC. \]

\[ d_{ij} : \text{euclidean distance between customer } i \text{ & } j. \]

\[ \text{MaxDis}_{ij} : \text{maximum length of a tour a vehicle can do.} \]

\[ y^k_{ij} : \text{current weight of vehicle } k \text{ while traversing from customer } i \text{ to } j. \]

\[ t^k_{ij} : \text{time consumed by the vehicle } k \text{ to traverse from customer } i \text{ to } j. \]

\[ [ET_i,LT_i] : \text{time window of package } i. \]

\[ ET_i : \text{earliest time at which a package } i \text{ can be delivered to its owner.} \]

\[ LT_i : \text{latest time at which a package } i \text{ is allowed to be delivered to its owner.} \]

\[ Sk_i : \text{service time needed to deliver the package to the customer } i \text{ by the vehicle } k. \]

\[ A_i : \text{arrival time to customer } i. \]

\[ D_i : \text{leaving time from the customer } i. \]

\[ FC_v : \text{fixed cost of the vehicle } v. \]

\[ VarC_v : \text{variable cost of a vehicle } v \text{ within a distance unit (ex: power supply cost).} \]

\[ x^m_{ij} = \begin{cases} 1, & \text{if a vehicle } m \text{ of a distribution center } k \text{ can travel directly from } i \text{ to } j. \\ 0, & \text{otherwise} \end{cases} \quad (1) \]

3.1 The Objective Function

1. Minimize the cost of the tour

\[ \sum_{k \in V_{m}} \sum_{i \in D \cup C} \sum_{C \in D \cup C} FC_k \cdot x_{ij}^m + \sum_{k \in V_{m}} \sum_{i \in D \cup C} \sum_{C \in D \cup C} d_{ij} \cdot x_{ij}^m \cdot VarC_k \quad (2) \]

Subject to these constraints:

(a) Maximum Distance Constraint: the length of the tour taken by a vehicle cannot exceed the maximum length of the tour that is specified for it (due to the fuel or energy supply)

\[ \sum_{j \in D \cup C} \sum_{i \in D \cup C} d_{ij} \cdot x_{ij}^m \leq \text{MaxDis}_{k}(k \in V_{DC}, m \in D) \quad (3) \]

(b) Capacity Constraints:

the load of any vehicle \( k \) traversing from a customer \( i \) to \( j \) cannot exceed the maximum capacity of that vehicle.

\[ 0 \leq y_{ij}^k \leq Q_k \quad (4) \]

(c) Time Constraints: each customer’s package \( i \) is usually attached with a time window \([ET_i,LT_i]\) that specify the time period in which the customer \( i \) is expecting to receive his package, not before the earliest time \( ET_i \) or after the latest time \( LT_i \). We can conclude with it the following information: The arrival time \( A_j \) to customer \( j \) from \( i \).

\[ A_j = D_i + t^k_{ij} \quad (5) \]

The departure time \( D_i \) from customer \( i \)

\[ D_i = A_i + S_i \quad (6) \]

We should attain the following constraints

\[ ET_i \leq A_i \leq LT_i \forall (i \in C, k \in V_{DC}) \quad (7) \]

\[ ET_i \leq S_i + A_i \leq LT_i \forall (i \in C, k \in V_{DC}) \quad (8) \]

(d) Precedence Constraints: in case of multi deposit scenarios, we need to guarantee that a customer will not be visited before his provider:

\[ D_i \leq D_j \quad (9) \]

(e) Each customer can be served by single vehicle only

\[ \sum_{m \in D \cup C} \sum_{i \in D \cup C} x_{ij}^m = 1 \forall j \in C \quad (10) \]

(f) The tour of each vehicle must start from a distribution center and finish at the same center.

\[ \sum_{j \in D \cup C} x_{ij}^m = 1 \forall (j, m \in C, k \in V_m) \quad (11) \]

(g) Path Controllability: every vehicle must leave a customer location only after reaching and serving that customer.

\[ \sum_{i \in D \cup C} x_{ij}^m = \sum_{j \in D \cup C} x_{ij}^m \forall (q \in D \cup C, m \in D, k \in V_m) \quad (12) \]

So, the final objective function is to minimize

\[ \alpha \cdot \sum_{k \in V_{m}} \sum_{i \in D \cup C} \sum_{C \in D \cup C} FC_k \cdot x_{ij}^m + \sum_{k \in V_{m}} \sum_{i \in D \cup C} \sum_{C \in D \cup C} d_{ij} \cdot x_{ij}^m \cdot VarC_k + \beta \cdot \sum_{m \in D \cup C} \sum_{i \in D \cup C} x_{ij}^m \cdot VarC_k \quad (13) \]

Where \( \alpha, \beta \) are weight constants to weightage the function’s value in the overall objective function.
4 PROPOSED MULTI-AGENT FRAMEWORK

Multi-agent approach provides speedup and computation efficiency as it supports exploring different areas of the search space simultaneously, producing a higher level of diversity and mature solutions (Czarnowski et al., 2013).

One of the important multi-agent paradigms is A-Team (Asynchronous Team) that was first proposed in (Talukdar et al., 2003) that describes a set of asynchronous independent software agents that communicate to jointly solve complex problems that can not be solved by single agent alone. A central memory stores a population of initial solutions that will be continuously modified by the software agents.

We will build the proposed distributed framework based on A-Team architecture. It consists of two types of agents, computing agents, each of which represents a logistic service provider and can run on any machine, and a coordinator agent, which is responsible for maintaining the central memory, and managing the communication between computing agents. Figure.2 shows the architecture of our proposed framework. Now we discuss the types of agents included.

4.1 Types of Agents

1. The Coordinator Agent:
   It is a single agent in the proposed architecture that manages a pool of initial valid solutions to the collaborative problem. It receives requests from every participating LSP agent to send it a subset of the available solutions to use it as initial solutions in its computation method. The coordinator agent is considered a synchronizing entity in the system since it manages the communication between different LSPs agents and is kept up to date about their latest status through message passing.

2. Logistic Service Provider Agent LSP:
   Each logistic provider is represented in the problem-solving process by an LSP agent that operates on their behalf. The LSP agents do not interact together directly; they exchange information about their state and found solutions through the coordinator agent. In this work, every LSP agent runs a meta-heuristic strategy that is different from the other agents and solves the given problem instance from its perspective and according to its personal priorities and goals, such as following potential SLA agreement. LSP agent finds the best solution and delivers it to the coordinator agent to be evaluated and possibly added to its pool of solutions replacing the worse existing solutions there. The coordinator may then resend the solution to other LSP agents in the next computation round in order to redirect their search to a potentially better region.

4.2 Communication between Agents

1. 2-way Handshake:
   Before starting the computations, every LSP agent registers itself with the coordinator to enable sending and receiving computation messages from the coordinator. This process starts with a SUBSCRIBE message sent by LSP agent to the coordinator, indicating that it is ready to participate in the computation and start its tasks. Then, the coordinator registers the LSP in the coalition and sends back a CONFIRM message to the LSP agent so it can be prepared to the next phase, the message contains the expected number of computation rounds that every LSP shall execute. Once the coordinator agent successfully receives the SUBSCRIBE message from all expected LSPs agents, the computation rounds start. Figure 3 shows the sequence of messages between the agents in the handshake phase.

2. Computation Messages:
   After all LSPs have registered with the coordinator, the coordinator initializes a collection of random feasible solutions to the collaborative problem. Periodically, every LSP agent sends a REQUEST message to the coordinator asking for a subset of the solutions from its pool, so the agent uses it as initial solutions in its solving method. The coordinator reply with an INFORM message containing a random subset of solutions. After receiving the initial solutions by the LSP agent,
it starts running its search strategy to find the best possible solution. When the search ends, the LSP agent sends the best solution as an INFORM message to the coordinator, which in turn compares the solution’s quality, in terms of overall cost-minimizing, to the existing solutions in the pool. If its quality is better, the new solution is added to the pool, replacing a less quality solution. Figure 4 shows messages exchange between the agents in the computation phase.

4.3 Search Strategies

Different search strategies for every agent are carried out in the proposed approach. Each LSP agent uses an adaptive large neighborhood search method (ALNS) as the underlying search method (Pisinger and Ropke, 2007). ALNS is a local search that is based on ruin and recreate principle that was proposed in (Schrimpf et al., 2000). It uses optimization techniques like threshold accepting and simulated annealing to partially destroy and repair the solutions. There are several strategies for the ruin phase, namely, radial ruin, random ruin, and cluster ruin (Pisinger and Ropke, 2007). A feasible solution is obtained by excluding a set of customers. In the recreate phase, all excluded customers are reintroduced to the solution wisely to minimize their effect on the complete solution. Some of the most widely used techniques are greedy and regret insertion methods.

4.4 Gain Assignment

In real-life collaboration, participants of any coalition mostly care about their profits and interests rather than the overall gain. Therefore it is crucial to estimate the individual gain for every participant in the coalition. In this paper, we will use two gain assignment methods. The first method is the Shapley value, which is a concept that was first presented in cooperative game theory as one way to distribute the total gain of a coalition \( N \) among the participants in a fair manner. It captures the marginal contribution made by a participant \( i \) over all permutation \( S \subseteq N \) (Ouhader and El Kyal, 2017).

\[
\phi_N(i) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} k \times (cost(S \cup i) - cost(S)) \quad (14)
\]

Another method is a proportional method, a demand-based proportional gain assignment that depends on the demands of the participant served by the coalition. Let \( cost(s) \) be the cost of the best solution found by given coalition \( s \). Then, the proportional assignment of the participate \( i \) is given as :

\[
\phi_i = \phi_i \times cost(s) \quad (15)
\]

where \( \phi_i \) is the demand-based factor.

We explore two collaborative scenarios. First, we consider a full collaboration level where the goal is to achieve the minimum cost possible. Table 1 shows the characteristics of participating LSPs and the costs of non-collaborative routing.

<table>
<thead>
<tr>
<th></th>
<th>No. of orders</th>
<th>No. of vehicles</th>
<th>Demands</th>
<th>Individual routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSP1</td>
<td>82</td>
<td>13</td>
<td>3591</td>
<td>2469.3</td>
</tr>
<tr>
<td>LSP2</td>
<td>90</td>
<td>13</td>
<td>4478</td>
<td>2644.4</td>
</tr>
<tr>
<td>LSP3</td>
<td>77</td>
<td>13</td>
<td>4037</td>
<td>2467.6</td>
</tr>
<tr>
<td>Total</td>
<td>249</td>
<td>39</td>
<td>12106</td>
<td>7581.3</td>
</tr>
</tbody>
</table>

Whereas in the second scenario, personalized collaboration is considered as every LSP agent works accord-
Table 2: Cost reduction in collaborative scenarios and the satisfaction level of the participants.

<table>
<thead>
<tr>
<th>Best solution</th>
<th>Overall reduction</th>
<th>Individual satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>No. of vehicle</td>
</tr>
<tr>
<td>Full collaboration</td>
<td>4142.1</td>
<td>26</td>
</tr>
<tr>
<td>Personalized collaboration</td>
<td>4588.9</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 3: The routing cost allocation of LSPs under collaborative scenarios.

<table>
<thead>
<tr>
<th>Best solution</th>
<th>Cost share / Shapley value</th>
<th>Proportional cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total cost</td>
<td>No. of vehicle</td>
</tr>
<tr>
<td>Full collaboration</td>
<td>4142.1</td>
<td>26</td>
</tr>
<tr>
<td>Personalized collaboration</td>
<td>4588.9</td>
<td>26</td>
</tr>
</tbody>
</table>

5 EXPERIMENTS AND RESULTS

To implement our proposed architecture, we used JADE framework (Java Agent DEvelopment), which supports FIPA specifications. We created a system consisting of one coordinator agent and three LSP agents representing three different logistics service providers in the industry. In our experiment, ALNS algorithm is used as the primary method, along with different ruin and recreate strategies for every LSP agent. LSP1 agent uses ALNS with random ruin strategy and regret/best insertion, LSP2 agent uses radial ruin strategy, and regret insertion, while LSP3 uses cluster ruin strategy and regret/best insertion. The coordinator agent maintains a pool of 20 initial solutions which will be continuously updated. We consider a multi-depot vehicle routing problem MDVRP; we use a dataset derived from famous Cordeau datasets (Cordeau et al., 1997) with few adjustments to fit the collaborative scenario. We will evaluate the proposed approach based on its performance in solving this problem.

Table 1 shows the characteristics of participating LSPs and the cost and vehicles needed for their individual routing, where every LSP uses its resources to serve its own customers only with no collaboration. Table 2 shows the cost of the best solution for collaborative routing scenarios using our proposed framework. We calculate the overall cost reduction of collective routing as the marginal percentage of \((Total Cost_{no, collaboration} - Total Cost_{collaboration}) / Total Cost_{no, collaboration} \times 100\). Thus, we found that full collaboration between LSPs reduced the total costs of a margin of 45% with an almost varied level of satisfaction for the participants in terms of the average index of delivered packages since the overall cost reduction was the only goal. In the second scenario, when considering personal goals and interests, a decrease in total savings is observed compared to the first scenario, along with an increase in participants’ satisfaction levels to almost equal value as every agent works to achieve its personal goal first. Table 3 shows the cost allocation for every participant LSP in both collaboration scenarios calculated by methods mentioned in Eq.14 and Eq.15 respectively. The overall results show that participants’ satisfaction levels can be increased by tolerating some loss in profits.

6 CONCLUSIONS

We have presented a multi-agent framework that supports the collaboration between different LSPs in the logistics level serving their customers jointly. The proposed framework is domain-independent and can be used for different cooperation scenarios in problem-solving. Furthermore, it is computationally efficient and scales to large size problems as it exploits exploring different areas of the search space simultaneously, and produce diverse and mature solutions. When the overall benefit is prioritized, some unintended biases to one of the participants may happen, and that will affect the trust in the coalition. Therefore, we examined when participants work to achieve objectives according to their preferences instead of focusing on the unified objective. We no-
ticed that collaboration could achieve a cost reduction of near 45% compared with individual routing, along with different satisfaction levels. However, when personalized goals are considered, the overall saving is comparatively reduced, but better levels of satisfaction are obtained. Therefore, considering the egalitarian approach that guarantees an equal level of service and satisfaction may encourage different LSPs to work towards further collaboration on several levels.

REFERENCES


