http://dx.doi.org/10.14257/ijca.2014.7.7.34

Design of Adaptive Vehicular Agents Model

Jin-Hong Kim[1], Hoon Jin[2] and Seung-Cheon Kim[1]†

[1]*Department of Computer Engineering, Hansung University*
[2]*Department of Computer Engineering, SungKyunKwan University*
[1]*Department of Information Communication Engineering, Hansung University*
[1]*jinhkm@hansung.ac.kr*, [2]*bioagent@gmail.com*, [1]*kimsc@hansung.ac.kr*

Abstract

Traffic system, in general, like ruralized/urbanized area expansion traffic or pedestrian signals, and footslogging crowds consist of many autonomous, intelligent entities, which are distributed over a large area and interact with each other to achieve certain goals. However, these entities many represent completely different things, like traffic lights, trucks or even road users. According to these entities, we aim at giving insight into the concept of adaptive vehicular agents. Additionally, different applications of multi-agent system to the traffic domain are presented. Furthermore, we propose to traffic flow models which describe driver's behavior not only on a reactive but also on a cognitive level.

Keywords: *Adaptive Vehicular Agents (AVA), Traffic Domain, Driver's Behavior*

1. Introduction

In recent years, the concept of an agent has become popular in many research areas, like information communication engineering, computer science, sociology, biology or even mechanical engineering and so on [1]. Depending on the application area the term agent can have many different meanings [2]. Although they are different with occasion to situation, environment or application domain, an agent define *five self-** as the bellows; *(1) self-Situation*. Every agent is situated in an environment. It perceives information via sensors and acts on the environment via effectors. The definition of the environment depends on the application. *(2) self-Reaction*. Changes in the environment are recognized by the agent and the behavior is modified in time. Therefore, their behavior is flexible, dependable; the actions are not predefined but depending on the situation. *(3) self-Autonomy*. This means that the agent is self-determining and decides about their actions only with regard to its perceptions and internal/external knowledge. *(4) self-Socialization*. The agent interacts and communities with other agents. The social contact is essential in order to coordinate behavior, achieve common plans or solve problems cooperatively. *(5) self-Rationalization*. Every agent is committed to its goals. This is very important since the behavior should be more than clear reaction to the environment. The agent should even be able to show an initiative because of an internal/external motivation. Basically, two conceptual kinds of agents can be identified: *reactive and cognitive* agents [3]. A reactive agent only maps possible perceptions to the available reactions, that it means stimulus-response systems [4]. Such simple behavior is often used in car-following models [5]. The cognitive or deliberate agent is endowed with reasoning capabilities. In order to model complicated behavior, it's helpful to distinguish different types of behavior. In the context of cognitive classes are shown; *(1) Skill-based, (2) Rule-based, (3) Knowledge-based*. Like this, this could be

ISSN: 2005-4297 IJCA

information gathered by previous experience and describe all goals of an agent, whereas the intentions are generated from reasoning about the current beliefs and goals [6]. In order words, our newly idea is to what we make well-defined specific model to five self-* by AVA.

2. Agent Scheduling Problems

2.1. Problem Definition

The traffic domain is full of complex systems, where the solution process can be based on agents since the objects can be naturally identified as agents, *e.g.*, transportation planning and scheduling or road traffic control. In many of these applications the question arises, how different autonomous entities can work together effectively to achieve a common goal. However, Vehicular Network-based has gradually developed and implemented an intelligent transportation system, which is the Ministry of Science and Technology, Ministry of Land, *e.g.*, in Korea. The characteristic of this vehicular service is as follows;

- *A modern ITS system and a Vehicular Network infrastructure with about AP operators receiving orders concurrently.*
- *Each with a GPS navigation system and CCTV.*
- *A large variety of vehicles, including some with special equipment (Smart Platform, e.g.) for drivers.*
- *Unpredictability of the traffic congestion in various parts of Korea.*
- *Deriver experience; place where drivers live and status.*

Scheduling of vehicles and drivers under such conditions represents an exceedingly complex process, which is not feasible to achieve with any known methodology. Especially, in urban areas where the road network is dense the dynamics are basically governed by the traffic lights. However, it has been found that in urban traffic the patterns, rush-hours, are quite recurrent. Therefore, a lot of traffic signal control methods are based on feedback algorithms (Genetic algorithm, Ant algorithm, Machine Learning, *e.g.*,) using historical traffic demand data [7]. But the effectiveness of traffic control system *should work in real-time, and should be autonomous, self-adaptive and proactive [8]. Additionally, there is an inherent distribution of functionality over the components.*

2.2. Adaptive Scheduling Approach for AVA Model

In order to improve this problem describe in a new approach to scheduling was developed, present problem-solving framework that explores job decomposition, optimal task allocation, and solution formulation. In addition, examples from various problem domains are used to illustrate each of these techniques.

2.2.1. Problem Solving Framework

The process of AVA planning and problem solving can be modeled by a 4-phase closed-loop command and control architecture [9], shown in Figure 1 as bellows.

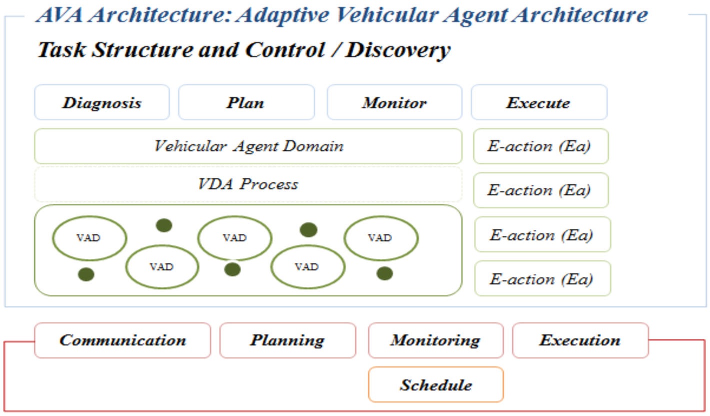

Figure 1. Problem Solving Framework is Represented by a Closed-loop Command and Control Architecture

This architecture contains the following sub-functions; Plan (P), Execute (E), Monitor (M) and Diagnose (D). The relation represents the flow of data between each of these functions, which forms a continuous feedback mechanism. Each of the functions in Figure 1 can be further decomposed to address the more specific problem of adaptive vehicular network planning and scheduling, as shown Figure 2. In this figure, the specific tasks of job decomposition, job/agent mapping, optimal schedule computation, task allocation, and solution formulation are represented within this higher level architecture.

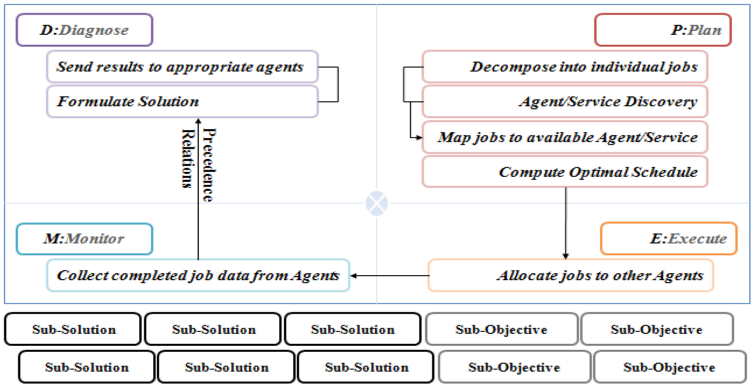

Figure 2. Agent Planning and Scheduling Process

As shown in Figure 2, these steps consist of ;

- Formulating or receiving from another source a global objective to be achieved.
- Structuring this objective in a form where it can be easily decomposed into a partially-ordered set of sub-problems or jobs.

- Surveying the environment for available agents and services that are capable of completing them.
- Determining the allocation of jobs to agents, such that the resulting schedule is optimized according to driver-oriented parameters.
- Formulating the solution based on results from all participating agents, and forwarding this solution to the appropriate agents.

2.2.2. Problem Solving Framework

In the AVA framework that we are considering, there is a set of heterogeneous 'avagent', so called 'agent' generally, in the vehicular network (V-network), each with a different set of available and services. A single agent is given an objective to complete, possibly from another agent, and it wants to take advantage of the resources provided by these other agent in the V-network to complete the objective more efficiently. Agent planning's first step is to decompose their objectives into a set of jobs, that is managed to task for planning on V-network, that could be allocated to other agents in the V-network and complete in parallel [10]. However there are often many possible problem formulations for a given objective and choosing the best method to decompose the objective may depend on the structure of the agent organization the number of different service types provided by these agents. In that case, our main goal is to choose the job precedence graph that produces the schedules with the most of interesting algorithm in which is famous 'Dijkstra's shortest path' and Genetic Algorithm in Figure 3 as bellows.

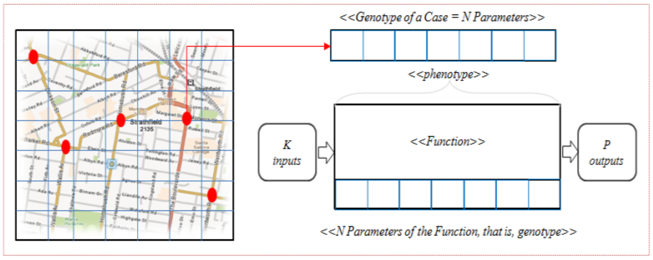
<<Genotype of a Case = N Parameters>>
<<phenotype>>
K
inputs
<<Function>>
P
outputs
<<N Parameters of the Function, that is, genotype>>
Strathfield
2135

Figure 3-A. ITS System Model based Genetic Algorithm

It is shown that is going destination from urban. (From Strathfield to Burwood in Australia). This shows a figure consisting of same red color of different each line, but we just consider 4-th location in order to example.

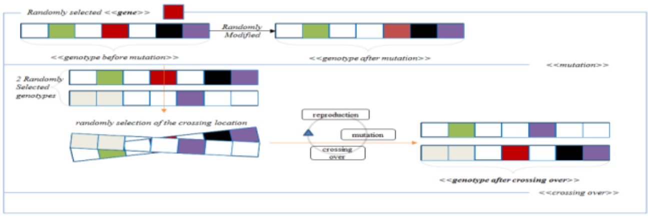
Randomly selected <<gene>>
Randomly
Modified
<<genotype before mutation>>
<<genotype after mutation>>
<<mutation>>
2 Randomly
Selected
genotypes
randomly selection of the crossing location
reproduction
mutation
crossing
over
<<genotype after crossing over>>
<<crossing over>>

Figure 3-B. This Case is shown that Phenotypes are Modified through 2 Ways;

Firstly, it is mutation that it consists in moving one parameter through a random modification (upper). Secondly, it is crossing over from the 2 randomly selected genotypes (lower).

2.2.3. Functional Decomposition

Large-scale problems from many different domains can be decomposed by defining them using generalized functions of the form {S(x), C(y), A(z)…} (S is Sensors, C is Control, and A is Actuators). A computer program or systems engineering problem may be describe this way where each function corresponds to an atomic operation or set of operations. These operations may be ordered relative to each other or may iterate through other operations recursively. As we mentioned above, we consider the following control system in the vehicular network (V-network).

2.2.4. Schedule Optimization and Task Allocation

After choosing an acceptable job composition and gathering all relevant information about the agent v-network, a problem-solving agent must determine which other agents to allocate tasks to. When computing an optimal schedule, an agent may be trying to achieve one or more of the following objectives;

- *Minimize the weighed completion time of each job*
- *Minimize the total number of agents used to solve the problem*
- *Minimize the amount of communication bandwidth used*
- *Maximize the aggregate value*
- *Minimize the aggregate cost*

3. V-Network based Adaptive Schedule with Algorithms

The distributed paradigm enables complex situations to be represented in the form most natural for users. It also enables software to be partitioned into relatively self-contained parts. It is compatible with the basic trend in modern optimization techniques, most of which are based now on a family of simultaneously competing and/or cooperating optimization algorithms. We newly approach the idea of Dijkstra’s algorithm;

Step 1. Maintain an estimate d[v] *of the length* δ(s, v) *of the shortest path for each vertex* v.

Step 2. Always d[v] ≥ δ(s, v) *and* d[v] *equals the length of a known path* (d[v] = ∞ if we have no paths so far).

Step 3. Initially d[s] = 0 *and all the other* d[v] *values are set to* ∞. *The algorithm will then process the vertices one by one in some order.*

Step 4. The processed vertex’s estimate will be validated as being real shortest distance, i.e. d[v] = ∞(s,v).

Step 5. Here "processing a vertex u*" means finding new path updating* d[v] *for all* v ∈ Adj[u] *if necessary. The process by which an estimate is updated is called* relaxation.

Step 6. When all vertices have been processed, d[v] = δ(s,v) *for all* v.

We consider an edge from a vertex u to v whose weights w(u,v). Suppose that we have already processed u so that we know d[u] = δ(s, u) and also computed a current estimate for d[v]. Then,

- There is a path from s to u with length d[u].
- There is a path from s to v with length d[v].

Combing this path from s to u with the edge (u,v), we obtain another path from s to v with length d[u] + w(u,v). If d[u]+w(u,v) < d[v], then we replace the old path <s,…,w,v> with the new shorter path <s,…, u,v>. Hence we update in Figure 4.

- d[v] = d[u] + w[u,v]
- pred[v] = u (originally, pred[u] == w)

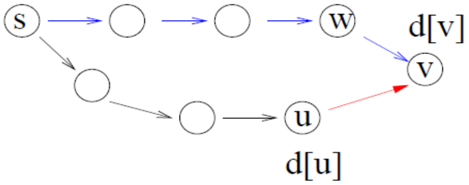

Figure 4. Shorter Path in V-network

We now prove that d[y] = δ(s,y). We have done relaxation when processing x, so

$$d[y] \leq d[x] + w(x,y) \tag{1}$$

Since x is added to S earlier, by hypothesis,

$$d[x] = \delta(s,x) \tag{2}$$

Since <s,…,x,y> is subpath of a shortest path, by (2)

$$\delta(s,y) = \delta(s,x) + w(x,y) = d[x] + w(x,y) \tag{3}$$

By (1) and (3), d[y] ≤ δ (s,y).

So y ≠ u (because we suppose d[u] > δ(s,y)). Now observe that since y appears midway on the path from s to u, and all subsequent edges are non-negative, we have δ(s,y) ≤ δ(s,u), and thus d[y] = δ(s,y) ≤ δ(s,u) < d[u].

Thus d[y] < d[u], which means y would have been added to S before u, in contradiction to our assumption that u is the next vertex to be added to S.

After all, all of the priority queue operations require O(log|Q|) = O(log n) time we have that the algorithms uses nO(1+log n)+O(e)+O(elong n) = O((n+e)log n) time.

4. Adaptive Vehicular Agent Discovery

In order to optimally solve a AVA scheduling problem using the algorithm presented in this research, it’s necessary to first have full information about the agent environment and the resources that are available to solve the problem. Therefore, a planning must discovery the structure of the physical network, including all services offered by various policy, their execution times, and communication delays before determining how to allocate jobs to AVA. Considerable research has been done on inter-AVA communication and discovery techniques that can help an agent to formulate a scheduling problem, we already had above, A pseudo-code representation of Dijkstra’s algorithm is shown bellows.

```
Initialize_single (Graph g, Node s) {
    for each node n in Nodes (g)
        g.distance [n] = infinity
        g.pi[n] = nil
        g.distance [s] = 0;
}

Relax (Node u, Node v, cost [ ][ ]) {
   if distance [v] > distance [u] + cost [u][v] then
        distance [v] = distance [u] + cost[u][v]
        pi[v] = u
}

Dijkstra (Graph g, Node s) {
   Initialize_single_source (g,s)
  S = {0}
  Q = Vertices (g) // priority queue
  While Q is not empty
     U = ExtractCheapest (Q);
     Add node u to S;
  For each vertex v adjacent to u relax (u,v,cost[u][v]) }
```

5. Conclusions

The goal of this research has been to explore AVA scheduling complexity and to develop a technique for solving the most difficult classes of these problems. A high level, closed-loop framework is used to model the various stages of AVA problem-solving, including problem formulation and decomposition, schedule computation, and task allocation. Extension to the work done in this research may include exploring a dynamic programming approach to the same problem that is capable of taking into account uncertainty levels or a limited view of the AVA environment, and can adapt to unexpected changes in the environment or problem domain during execution of the actual schedule. Other future areas of research related to this research may also include dynamically re-organizing of AVA that response to changing objectives and environments.

Acknowledgement

This research was financially supported by Hansung University.

References

[1] M. Molina, J. Hernandez, J. Cuena, "A Structure of Problem-solving Methods for Real-time Decision Support in Traffic Control", International Journal of Human and Computer Studies (Academic Press) N.49, 577-600, **(1998)**.

[2] J.-H. Kim, S.-C. Kim, "Design of Architectural Smart Vehicle Middleware" Information: An International Interdisciplinary Journal, Volume 16, Number 4, April, **(2013)**.

[3] Kohler, Walter H. "A Preliminary Evalution of the Critical Path Method for Scheduling Tasks on Multiprocessor Systems", IEEE Trans. On Computers, vol. 24, no. 12, pp. 1235-1238, (1975).

[4] Kwok, Yu-Kwong and Ishfaq, Ahmad. "Dynamic Critical-Path Scheduling: An Effectuve Technique for Allocating Task graphs to Multiprocessors". IEEE Trans. On Parallel and Distributed Systems, Vol. 7, p. 506, 196.

[5] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "TBD: Trajectory-Based Data forwarding for Light-Traffic Vehicular Network," Tech. Rep. 08-040, **(2008)**.

[6] Jin-Hong Kim, Seung-Cheon Kim, "Toward Hybrid Model for Architecture-oriented Semantic Schema of Self-adaptive System" International Conference on Green and Human Information Technology (ICGHIT), LNCS 7861, **(2013)**.

[7] Spears, W. and DeJong, K. "An Analysis of Multi-Point Crossover. Foundations of Genetic Algorithms", G. Rawlins, ed. Morgan-Kaufmann,. **(1991)**

[8] T. Starkweather, D Whitley and K. Mathias, "Optimization using Distributed Genetic Algorithms. Parallel Problem Solving from Nature, Springer-Verlag. **(1991)**

[9] M. Salehie and L.Tahvildari, "Software Technologies Applied Research (STAR) Group, University of Waterloo", Waterloo, Canada, ACM Transactions on Autonomous and Adaptive Systems,

[10] J. H. Kim, E.-S. Lee, KSVTs: Towards Knowledge-based Self-Adaptive VehiculeTrajectory Service, Information Technolgy Convergence, Lecture Notes in Electrical Engineering vol. 253, **(2013)**, pp.387-393.

Author

Jinhong Kim, he is Associate Professor of Department of Computer Engineering at the Hansung University, Seoul, and Republic of Korea. He respectively received his Ph.D. degrees in Computer Engineering from Sungkyunkwan University (SKKU), Korea, in 2006. Dr. Kim was a post doctorate research associate of the Software Engineering at the SKKU, as well as of the Information System Research Center at the University of New South Wales in Australia. Moreover, He acted as an Engineer expert on behalf of Vodafone Company. He has served on a Co-chair of the IEIE Computer Society. He also served or currently serving as a reviewer and Technical Program Committee for many important Journals, Conferences, Symposiums, Workshop in Computer Communications Networks area. His research interests include smart vehicular network, smart platform, software engineering, wireless sensor networks, scalable reliable communication protocols, mobile computing, network security protocols, proxy caching systems, and formal verification of communication protocols.