An environment friendly method to generate dynamic transportation routing in a distributed context

Da Xu, Bernard Archimède, Muhammad Ali Memon

An environment friendly method to generate dynamic transportation routing in a distributed context

(Presented at the 6th IESM Conference, October 2015, Seville, Spain) © I^{4}e^2 2015

Da XU, Bernard Archimède
Laboratoire Génie de Production ENIT
Tarbes, France
dxu@enit.fr Bernard.Archimede@enit.fr

Muhammad Ali Memon
Szabist
Hyderabad, Pakistan
drmuhammad.ali@hyd.szabist.edu.pk

***Abstract*—Due to the globalization, today's manufacturing industry has to mobilize as many as resources in order to satisfy needs of every customer. At the same time, the production resources are more often geographically distributed. In this context, the necessary shifting of intermediate products as well as the final delivery requires transportation among different manufacturing sites, warehouses, consumers, etc. As the business of enterprises grows, transportation demands also grow with that, resulting problems of increasing transportation cost, environmental pollution and coordination between production and transportation operations. In order to address these problems, a possible solution is to better define the routing for transportation requests. This paper proposes an environment friendly method based on a reduced weighted graph with transportation resources competencies to generate dynamic routing between pickup and delivery locations according to shortest distance and time in order to facilitate consolidation and consequently cost and pollution reduction**

Keywords—Transportation routing; Multi-agent System; Transportation planning; Interoperability; Collaborative networks

I. Introduction

Nowadays, manufacturers have to deal with a variety of competitors, suppliers and customers coming from several markets in the world. With the help of Information Communication Technology (ICT), suppliers, manufacturers, distributors, and retailers can share information rapidly and precisely [1]. To deal with the complexity of business routines, people use some automatic mechanisms to handle the data and information during the operation in logistics and manufacturing activities. Indeed, firms around the world have been implementing Enterprise Resource Planning (ERP) systems since the 1990s in order to have a uniform information system in their respective organizations and to re-engineer their business processes [2]. However, it remains the lack of better coordination methods between production scheduling, logistics and warehousing, etc.

In most of the transportation approaches, routes of the vehicle come from an optimization scheduling process based on pick up and delivery dates of the considered set of transport orders. In our context, we consider that vehicle's route is already defined by the transporter, on the basis of any criteria, for example future demand forecast. Example of this transportation is like train, bus or airplane routes whose circulation is predefined. Therefore, transportation planning assumed here has to optimally place transport orders in the vehicle's route in efficient way and routing for finding optimal path for each transport order is the pre-step of the planning. This routing may determine route comprising of delivery of transport order by one or more vehicles in transshipment. Moreover, each transport order may request routing on the basis of several criteria (shortest distance, time, cost, early deliver or trade of between two or more of these criteria). The goal of the paper is to define the routing in an efficient way, which eventually focuses on both transport order's priority criteria and efficient utilization of the transport resources in a collaborative context.

Driven by the requirement of dispatching distributed resources and high-performance collaboration between them, different multi-agent models have been proposed and applied trying to solve such scheduling tasks. Many researchers have proposed different active models to solve such distributed resource scheduling and planning problems. They proposed multi-agent system (MAS), which is a computerized system composed of multiple interacting intelligent agents within an environment [3]. It is designed to be competent to solve complex problems that are difficult or impossible for an individual agent or a monolithic system to solve by co-operation between several autonomous agents via collaborative networks.

In the context of transportation, whose objective is to support production, we consider transportation orders generating during the production process as well as for the final delivery. During a production process, one manufacturing site, usually a factory or workshop require delivery of intermediate products to the other sites for a next process. During the delivery, final products need to be shipped to warehouse or delivered to the customers. In both cases, a transport request arrives with basic information elements indicating starting location, delivery location, delivery time and etc. In order to plan those deliveries optimally, we need to find out the best path from the start to the destination. For the delivery of transport orders common constraint that arises is the limited transport resources (vehicles) usually operated by a single transporter service provider. In order to ensure the order's delivery, transportation service providers need to cooperate with each other. In this paper, we propose an autonomous Path Finder agent as a web service. It has an up-to-date awareness of all transportation resources, the combining network including all transporters. Before the planning, Path Finder will suggest a shortest path for each transportation order requested. In section 2, we present a state of the art literature on transportation planning methodologies and path finding technologies. Section 3 is dedicated explanation of Path Finder agent and the system proposed. In the section 4, we illustrate how Path Finder calculates the shortest path using a graph and how Path Finder finally works with a multi-agent system for facilitating transportation planning. Finally, we describe in section 4, the implementation of Path Finder and show the results with benchmarks.

II. State of art

A. *Transportation scheduling methodologies*

Two famous approaches for transportation planning solutions are operational research approaches and mathematical programming. In operational research approaches, a complete structure of the shops, activities, jobs, and all other related constraints can be represented in some detail as that, given appropriate input data and simple heuristic dispatching rules at decision points, the computer could extrapolate a given schedule into the future at a relatively low cost. Investigators should generate various types of artificial or historical input data and simulate the effect of using different types of simple heuristics under different conditions. In mathematical programming approach, linear integer programming and dynamic programming are the most widely mathematical techniques used for solving production-scheduling problems. The main advantage of such mathematical models is their capabilities of determining the optimal schedule. For large-scale problems, the mathematical formulation remains difficult and in some cases it cannot be derived [4].

The traditional scheduling methods encounter great difficulties when they are applied to real-world situations. This is because they use simplified theoretical models and all computations are carried out in a central computing manner. The multi-agent systems, on the other hand, suggest an innovative, lightweight approach to scheduling problems. Agent-based systems have been hailed as a new paradigm for conceptualizing, designing, and implementing software systems. This essentially distributed approach is more flexible, efficient, and adaptable to real-world dynamic manufacturing environments.

Recently, researchers have applied multi-agent systems trying to solve scheduling problems. Applications include manufacturing shop scheduling, transportation scheduling, construction scheduling [5], power distribution scheduling, Grid-scheduling [6], meeting and course timetable scheduling [7], medical test scheduling, and project management. Agent-based approaches have several potential advantages for distributed manufacturing scheduling [8]. These approaches use parallel and distributed computation through a large number of processors, which may provide scheduling systems with high robustness and efficiency.

B. *Path finding technologies*

The paper written by Leonhard Euler on the Seven Bridges of Königsberg and published in 1736 is regarded as the first paper in the history of graph theory [9]. In real life and production, many problems can be abstracted to models containing a graph e.g. social relationships, highway system. In the last decades, with the prosperity of operational research and computer science, graph theory has been enriched and applied in various fields. Recently, driven by more complex and combined scenarios, people have proposed ITS (Intelligent Transport System) [10] trying to fulfill more advanced transport criteria. But the methods in ITS are quite complex and focus too much on traffic control, navigation and low-level implementation. Dealing with transportation planning or logistics problems, we need a good abstraction model and there are always classic and common scenarios where similar mapping can be found in graph theory. In path finding or path planning, shortest path problem is one of these.

In this paper, our concern is finding the shortest path from one location to another in a transportation network in order to facilitate the scheduling of transportation and production planning. Typically, the single-source shortest-path problem was discovered by Dijkstra (1959), while the all-pair shortest-path algorithm is due to Floyd (1962), who obtained the result based on a theorem by Warshall (1962) [11]. When searching a shortest path, especially in graph with sparse nodes, Dijkstra is a most classical and mature algorithm [12]. When dealing with a dense graph, a Goal-Directed approach will be better e.g. A * search, Geometric Containers, Precomputed Cluster Distances [13]. In this paper, we develop our algorithms based upon Dijkstra and A * with some improvement.

III. Path Finder and system overview

In a traditional production scenario illustrated in Fig 3.1, we have several distributed manufacturing sites. Each site could be an enterprise or a workshop. Now these manufacturing sites have intention to combine to form a virtual enterprise to be more productive. But their manufacturing systems work separately. In order to manufacture products collaboratively, these sites need to exchange some intermediate products to form final products. In this case, they have to plan that production process in a distributed fashion. However, the question is how we transfer goods among distributed sites with respect to various manufacturing processes and budget optimization. To handle such a dynamic and collaborative scenario, then our model is evolved a generic model with

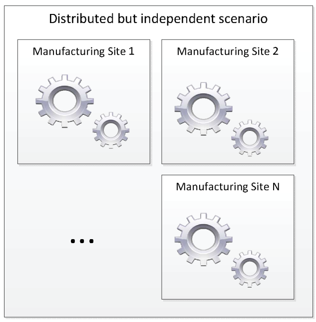

Fig. 3.1. Distributed manufacturing sites with their planning systems.

transportation planning in Fig 3.2.

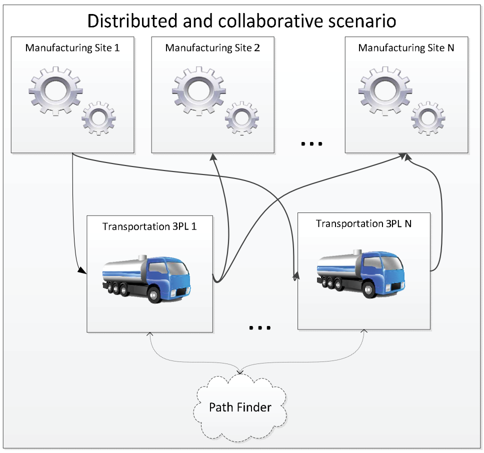

Fig. 3.2. Distributed manufacturing sites with transportation 3PL (3rd party logistics [16]) incorporated

As illustrated in Fig.3.2, we have new components, compared to Fig.3.1, that are responsible for transportation and Path Finder agent. Before transportation scheduling runs, each 3PL communicates with Path Finder agent to obtain the routing for each order. To fulfill the collaborative scheduling that covers a bunch of sites over a region, Path Finder agent must function in interoperable manner with every connected 3PL. Path Finder collects necessary information from these 3PL to build a real-time transportation network that governs all the public transportation resources. Whenever some manufacturing site proposes a request as a transportation order, Path Finder dynamically calculates a shortest path for this order. Path Finder keeps an up-to-date awareness of all the 3PLs' vehicles' availability and their effective displacement. We define each displacement as a Transport Activity (TA). For the geographic information like distance and duration of each TA, Path Finder will acquire freely from available Internet services like Google Maps. Next section is dedicated to explanation of shortest path and the system's architecture is presented in section 4.

IV. Implementation and Result

A. Distributed multi-agent scheduling system

Fig 4.1 illustrates a general architecture of the distributed multi-agent scheduling system including manufacturing sites and transportation service provider (3PL) as well as Path Finder agent. On the left side of Fig 4.1, both manufacturing and transportation sub-systems are physically distributed, while the multi-agent scheduling control unit is deployed in a centralized pattern shown on the right side (server side). When the scheduling request from manufacturing sites or transportation 3PL arrives at the master service who is in charge of supervising and collaborating the overall scheduling, the master service will allocate an exclusive server process responsible for all related operations of the scheduling request session. About how exactly each agent acts inside each sub-system and how they cooperate, please refer to [14] [15]. In Fig.4.2, each time a transportation scheduling request is sent to its exclusive server (T), Path Finder agent firstly returns a reasonable transportation routing that the scheduling must follow afterwards. Routing for manufacturing site means the proceeding sequence shifting between different machines or workshops. The similar process mapped to transportation is the sequence of cities and routes. Any transport requirement from any manufacturing site equals to a transportation order for a transportation 3PL (3rd party logistics).

In Fig 4.1, there is a component called Shared Resources Register. This component is especially important for Path Finder as it keeps updating shared transport resources registry so that the Path Finder is able to utilize these resources. For example, when transportation 3PL 1 wants to add its new purchased truck as a new public transportation resources in the system. It should notify Shared Resources Register the capability of the very truck and it competence i.e. how much the capacity of the vehicle is, from which location to where it circulates, what kind of timetable it follows each day, etc.

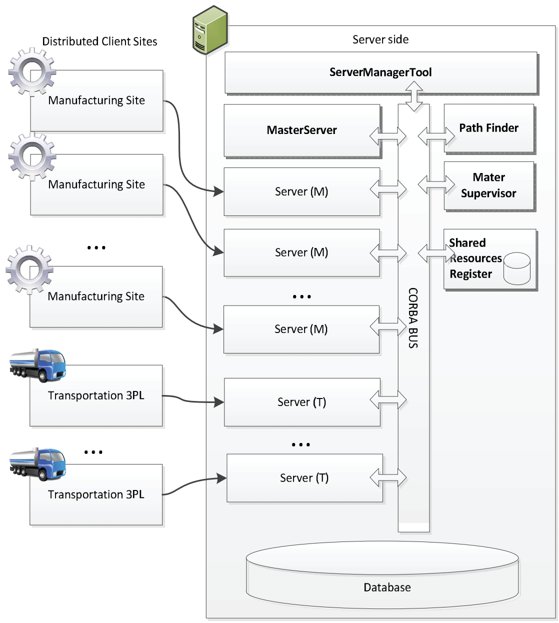

Fig. 4.1. Global architecture of multi-agent scheduling system integrated with Path Finder

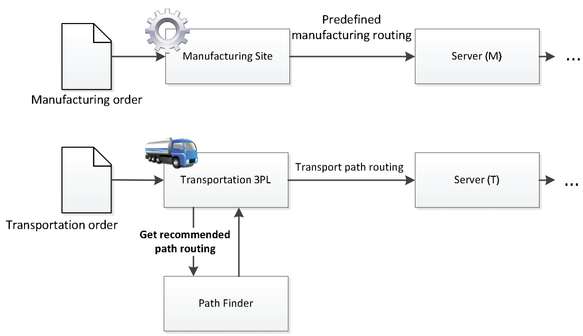

Fig. 4.2. Different dynamics when dealing with different order requests.

B. *Path Finder agent*

As a web service, Path Finder follows the basic design pattern of request and response between client and server. Fig 4.3 is a trimmed and simplified structure of Path Finder. Basically, there are 3 kinds of actions or behaviors defined for Path Finder: maintenance operations or status query, path query, and geographic data update query. Path Finder connection threads are the temporary sub-server for each request. Transport net context is the core graph containing up-to-data transport resources information. When a shortest path query request arrives, Path Finder will choose a proper algorithm according to the search strategy to search the transportation graph i.e. transport net context. When some new operating zone or TA are defined, Path Finder need to update their geographic information. Simultaneously, map service "querier" will access public map service to download and parse the newest geographic data, then store them into the shared resources register. Every request will follow certain standard format protocol recognized by Path Finder. Path Finder connection thread will parse the message and execute respectively.

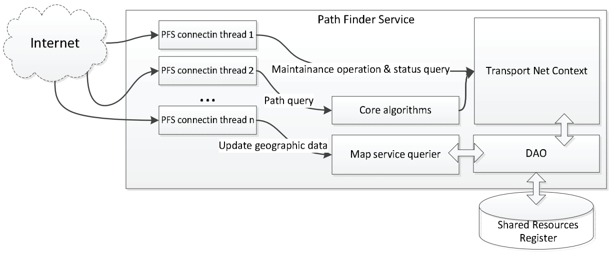

Fig. 4.3. How Path Finder agent deal with various request from client sites.

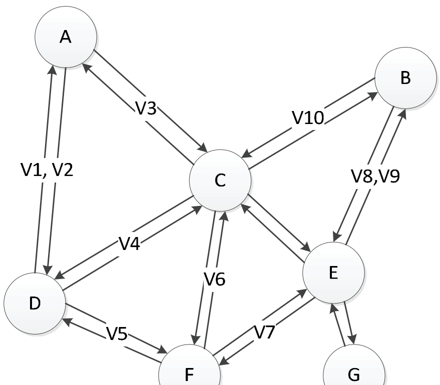

Fig. 4.4. A small example of directed graph representing the transportation network.

In the transport graph i.e. transport net context, we let each identified location to be a node and each TA (Transportation Activity) to be the directed arc between two nodes. In Fig 4.4 we take a small example in which we have locations labelling from A to G and vehicles from V1 to V10. Each location represents a manufacturing site, a transportation 3PL or a warehouse. Each transportation 3PL define their own TAs and vehicles as public transportation resources. Path Finder will collect information, which will be maintained in shared resources registry. Thus in the same graph, vehicles could be from different participants. While they work together in the same collaborative networks to fulfill scheduling from a global point of view. The directed arcs representing TAs usually appear in pair between two locations, which means that the vehicle will circulate back and forth instead of a one-way trip. The natural distance between two locations will be retrieved by public map service e.g. Google Maps. In fact, we only query the geographic distance and general transport time from public map service for those TAs that are predefined by 3PLs.

When the best path is about the shortest transportation time, notice that from A to D we have multiple available vehicles V1 and V2, then the transportation time from A and D will be the average value of V1 and V2. An important constraint is that the accessibility between locations must comply with vehicles' competences (the competences of a vehicle indicates those TAs it circulates), e.g. in Fig 4.4, even though we have arcs between C to E and E to G, but there is no any vehicle that will execute these TAs (we assume that because of accident or any other reason, some vehicles are temporarily unavailable), then between C to E and E to G are still blocked. Thus, we declare that the graph on which Path Finder acts is actually a reduced graph weighted with transportation resources competencies. This is also another reason why we have to make Path Finder to be a service capable of dynamically updating. Let's consider location G. Apparently, the only TAs that link G with other locations lack vehicle competences, thus G is absolutely isolated. Then Path Finder will response to any shortest path request whose destiny is G that the path is blocked.

C. *Algorithms performance analysis*

Several case studies were developed. The case study we choose in this section covers more than 60 metropolises located in France, part of Germany and Italy. An easy interface as shown in Fig 4.5 is also built to show the path graphically. A performance analysis comparing Dijkstra and A* algorithm

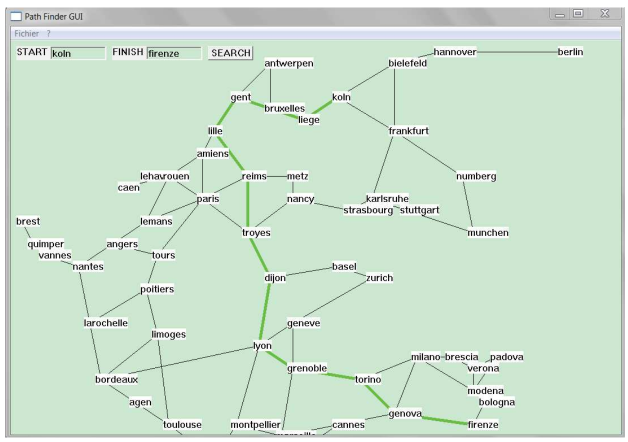

Fig. 4.5. An easy interface to show the path finding result

when dealing with shortest distance path query is shown in

Table 4.1 and Fig 4.6. In last columns of the table 4.1, we can see that, when the query strategy is BEST_DISTANCE i.e. shortest distance, A* algorithm is somewhat better than Dijkstra as predicted before. While on the other hand, Dijkstra seems to be more stable when the traffic graph is not too big. The other testing results show that Path Finder owns a quite acceptable performance outcome. Both the BFS (Breadth First Search) method applied to test if start and finish locations are reachable and the core algorithm have a response time less than 0.01 seconds. However, further tests and improvement are still needed especially when the graph grows to a bigger scale and in real network environment.

Table 4.1. Dijkstra and A* running results comparation in strategy BEST_DISTANCE

START	FINISH	LOOPS IN BFS	TIME Dijkstra (*ms*)	TIME A* (*ms*)
Vanes	Paris	14	3	4
Troyes	Munchen	92	5	2
Nantes	Toulouse	30	4	3
Lyon	Bologna	78	4	3
Padova	Bordeaux	32	4	2
Antwerpen	Limoges	65	4	4
Rouen	Numberg	110	6	4
Firenze	Berlin	289	6	7
Montpellier	Lille	56	6	2
Koln	Zurich	91	5	5
Tours	Frankfurt	94	5	3
Toulon	Caen	108	5	5

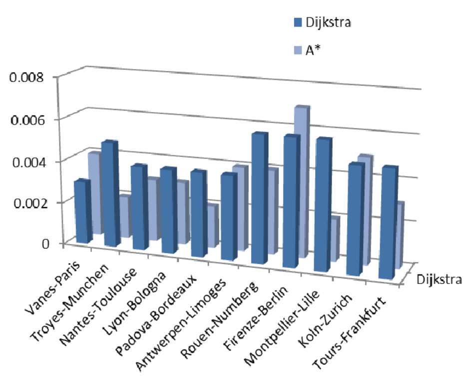

Fig. 4.6. An easy performance comparison of Dijkstra and A * under BEST_DISTANCE strategy.

Conclusion

For the delivery of transport orders common constraint that arises is the limited transport resources (vehicles) usually operated by a single transporter service provider. In order to ensure the order's delivery, transportation service providers need to cooperate with each other. In order to facilitate a distributed and collaborative transportation planning, we proposed a Path Finder agent as a web service in order to yield the best path for the delivery of transport orders between two production sites and final delivery to customers. Path Finder agent is designed to find out the routing of vehicles based on the different criteria like shortest distance or time between origin and destination. It builds and maintains a reduced graph weighted with transportation resources competencies. The geographic information (time and distance between two locations) that Path Finder uses to build the transport network graph is retrieved from public map service. Path Finder uses algorithms based on Dijkstra to find the shortest route. When we can find a reasonable heuristic function and the transport network is thin, we consider applying A* algorithm or other Goal-Directed methods which would be more efficient in that circumstance.

References

[1] Yu Li. Impact of modern logistics on industrial location choice and property markets. Thesis (Ph. D.), Massachusetts Institute of Technology, Dept. of Urban Studies and Planning. 2007.

[2] Rajagopal, P., 2002. An innovation-diffusion view of implementation of enterprise resource planning (ERP) systems and development of a research model. Information & Management 40, 87–114.

[3] Peter Stone, Manuela Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. Autonomous Robots, 8, 345-383, 2000.

[4] Hamdy Elwany, Mohamed Shouman, Mohamed Abou-Ali. Production Scheduling Techniques – A Review.

[5] T.Horenburg, J.Wimmer, and W.A.Günthner. (2012). Resource Allocation in Construction Scheduling based on MultiAgent Negotiation. 14th International Conference on Computing in Civil and Building Engineering.

[6] Mauricio Solar, Jorge Rojas, Marcelo Mendoza, Raúl Monge. (2012). A Multiagent-Based Approach to the Grid-Scheduling Problem. CLEIej 15(2).

[7] Mihaela Oprea. (2006). Multi-Agent System for University Course Timetable Scheduling. The 1st International Conference on Virtual Learning, ICVL. 231-238.

[8] Shen W. Distributed manufacturing scheduling using intelligent agents [J]. Intelligent Systems, IEEE, 2002, 17(1): 88-94.

[9] Biggs, N.; Lloyd, E. and Wilson, R. (1986), Graph Theory, 1736-1936, Oxford University Press.

[10] Kumari, S. Meena, and N. Geethanjali. "A survey on shortest path routing algorithms for public transport travel." Global Journal of Computer Science and Technology 9.5 (2010): 73-76.

[11] Algirdas Sokas. (2011). Algorithms and Procedures of Determining the Shortest Route in the Graph. TRANSBALTICA the 7th International Conference.

[12] Fan, DongKai, and Ping Shi. "Improvement of Dijkstra's algorithm and its application in route planning." Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on. Vol. 4. IEEE, 2010.

[13] Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P. & Werneck, R. (2014). Route planning in transportation networks. In Technical Report MSR-TR-2014-4. Microsoft Research, Microsoft Corporation.

[14] Memon, Muhammad Ali, and Bernard Archimède. "A Multi-Agent Distributed Framework for Collaborative Transportation Planning." In Manufacturing Modelling, Management, and Control, vol. 7, no. 1, pp. 1370-1375. 2013.

[15] Memon, Muhammad Ali, and Bernard Archimède. "Towards a distributed framework for transportation planning: A food supply chain case study." In Networking, Sensing and Control (ICNSC), 2013 10th IEEE International Conference on, pp. 603-608. IEEE, 2013.

[16] Marasco, Alessandra. "Third-party logistics: A literature review." International Journal of Production Economics 113.1 (2008): 127-147.