

# Agent-based Shipment Algorithm for Capacitated Vehicle Routing Problem with Load Balancing

Seog-Chan Oh<sup>1</sup> · Shang-Tae Yee<sup>2</sup> · Taioun Kim<sup>3†</sup>

<sup>1</sup>The Penn State University, University Park, PA 16802, USA

<sup>2</sup>General Motors R&D Center, Warren, MI 48088, USA

<sup>3</sup>Department of Industrial Engineering, Kyungsoo University, Busan 608-736, Korea

## CVRP를 위한 에이전트 기반 Shipment 알고리즘 개발

오석찬<sup>1</sup> · 이상태<sup>2</sup> · 김태운<sup>3</sup>

<sup>1</sup>The Penn State University / <sup>2</sup>General Motors R&D Center / <sup>3</sup>경성대학교 산업공학과

Load building is an important step to make the delivery supply chain efficient. We present a family of load makeup algorithms using market based control strategy, named *LoadMarket*, in order to build efficient loads where each load consists of a certain number of finished products having destinations. LoadMarket adopts Clark-Wright algorithm for generating initial endowment for *Load Traders* who cooperate to minimize either total travel distance or the variance with respect to the travel distances of loads by means of the spot market or double-sided auction market mechanism. The efficiency of the LoadMarket algorithms is illustrated using simulation based experiments.

**Keywords:** CVRP, Load Balancing, Double-Auction, Spot-Market Auction, Market based Controls

## 1. Introduction

Load building is a process of assigning a set of products to a number of transportation carriers like truck or railcar. Companies have made an effort to achieve more efficient shipment planning and execution of finished products on their delivery supply chain. The efficiency in shipment planning and execution can improve customer fulfillment with fast delivery and reduce total transportation cost. We present LoadMarket, a family of load assignment methods using market based control mechanism.

Let us consider a motivation example where the optimization of load assignment is crucial issue. Suppose an automotive company produces cars and transports them to the customers nationwide (Yee, 2002). As shown in <Figure 1>, whenever an ordered

car is produced in the plant, it moves to the yard where produced cars wait to be assigned to a truck that is the transportation carrier. Holding cars in the yard should be considered as a factor causing costs. Each truck has a capacity limit which is the maximum number of vehicles to carry at once and after containing one load of cars, it starts to visit each city to deliver cars. The transportation cost of each truck is proportional to the total travel distance. In summary, the daily cost model for this scenario can be as follows :

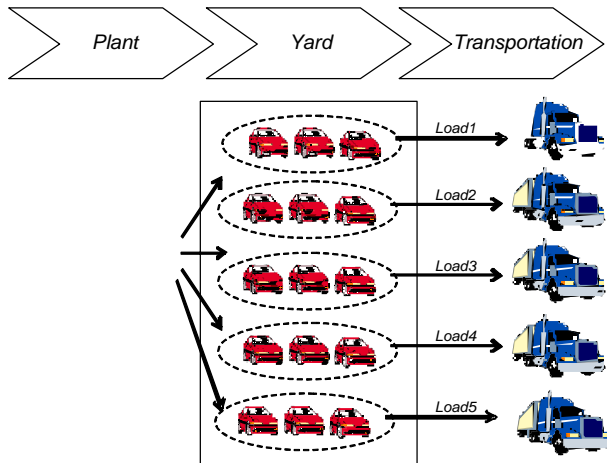
Cost model = (average # of cars in the yard maintenance fee per a day) + (daily # of trucks departing from the yard average travel distance of trucks)

The above cost model implies that there can be several cost saving strategies related with the typical

† Corresponding Author : Professor Taioun Kim, Department of Industrial Engineering, Kyungsoo University, Busan 608-736, Korea,

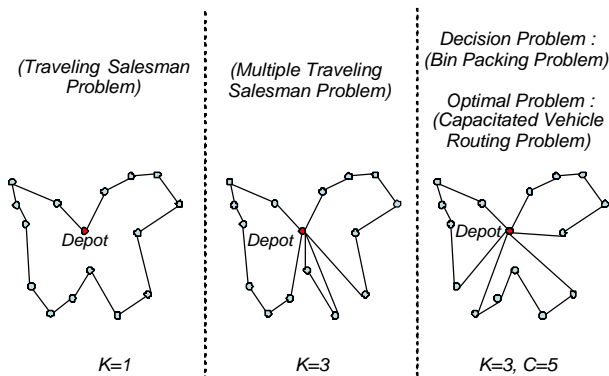
Fax : +82-51-621-2454, E-mail : twkim@ks.ac.kr

delivery supply chain example : (1) fast shipping out cars waiting in the yard and (2) finding a shortest average travel distance for each trucks. In this paper, we consider the second problem, which is finding the shortest average travel distance considering that travel distances between trucks, is required to be balanced as uniformly as possible.



**Figure 1.** A scenario of the load assignment.

For the typical vehicle routing problem (VRP) the goal is to determine vehicle route in order to minimize the transportation costs subject to meeting the demands of all delivery points. As VRP is known to be NP-hard and it needs a lot of computing time to get the optimal solution, many heuristic approaches has been proposed (Ro and Ye, 1996). For a revised solution to find a heuristic algorithm, a parallel genetic algorithm is applied (Yoo and Ro, 1999). Recently, integrating reverse flows into their transportation system draws much interest. Genetic algorithm is applied for the VRP with mixed delivery and pick-up (Chung and Park, 2004).



$K$  : Number of delivery trucks (loads)  $C$  : Capacity per a truck (load)

**Figure 2.** Relaxed problems of CVRP.

If there is a large truck with the ability to carry all

daily produced cars and cover all destination cities, this problem is reduced to a traveling salesman problem (TSP) (i.e., finding a path through a weighted graph which starts and ends at the same vertex) (Flood, 1956). However, in the real world, a truck can carry at most ten cars at one shipment and can travel at most over a few numbers of neighboring cities. Thus, the real scheduling problem involves multiple trucks in order to deliver large size of daily-based produced cars, which leads to the capacitated vehicle routing problem (CVRP) (Clark and Wright, 1964). For solving CVRP, we can start with handling two relaxed forms of CVRP at a high level due to the intractable complexity of CVRP as shown in <Figure 2>. The two relaxed forms, which are (1) Multiple Traveling Salesman Problem and (2) Bin Packing Problem will be discussed in next Section.

The presented example scenario often occurs in many other industries that aim to *Order to Delivery* (OTD) service. However, identifying minimized cost value when the number of cities to visit and size of items to deliver increase, is a challenging task, since in real application, such numbers can be many and non-trivial. Therefore, there is an imminent need for the method that systematically and mechanically helps to plan a load assignment such that average travel distance for delivering the loads is minimized and moreover, the load balancing between the loads is satisfactory.

In addition to the inherent complexity of CVRP with load balancing, there are other obstacles that contribute to increasing the problem complexity when it comes to thinking about the real practice as below. First is the less assured information. It is not unusual to change completed delivery schedules because of less assured information. For example, vehicles waiting for delivery in the yard might be held in a sudden without shipping when some serious quality problems happen. Therefore, once a delivery schedule has been formed, the schedule is likely to be revised frequently. In this case, rather than global optimal solution, people are generally satisfied even though the new schedule ends up being locally optimized. Second obstacle is the unstable truck arrival and departure time. It is common to outsource multiple truck companies to deliver vehicles with less expense. However, it causes problems such as unexpected truck arrival or departure time delay. Whenever these kinds of problems happen, delivery schedules are changed rapidly such that obsolete vehicles are not generated and OTD rate is not reduced.

Consequently, in addition to the intractable underlying mathematical scheme, these coupled hardness caused by less assured information and unstable operation times make it hard to solve the CVRP with

load balancing just relying on the traditional off-the-shelf optimization aimed techniques. Therefore in this paper, rather than the centralized and fully-informed optimization approaches, we suggest a family of algorithms based on intelligent agents in a distributed manner with auctioning for products (in this case, identical to customers) by the load traders.

Another motivation to use agent-based approaches in this paper can be found from the implementation perspective. The uncertain situation above typically requires streaming-like requests rather than elastic requests so that proper decision systems must be implemented in such a manner that they can generate the best feasible solution like anytime algorithm even though the solutions are not optimal. For showing the validity of our proposal, we compared ours with the most widely known heuristic used to solve the VRP, Clark-Wright method in the last Section. Our contribution can be summarized as follows :

- (1) We introduce the CVRP with load balancing problem and discuss its inherently intractable complexity and real delivery scheduling environment pressed by unexpected events.
- (2) We propose two market based algorithms that are naïve and blind due to the distributed manner but fully support the underlying auction mechanism.
- (3) We validate our proposals by conducting experiments through the comparisons with an existing algorithm. Through the validation, we prove that both market based algorithms overrun Clark-Wright method. In addition, we discuss pros and cons of both algorithms which arise because the strength of each algorithm varies depending on different performance measures.

## 2. CVRP with load balancing

More formally, we consider the CVRP with load balancing in which a quantity  $d_i$  of a single commodity is to be delivered to each customer  $i \in N = \{1, \dots, n\}$  from a central depot  $\{0\}$  using  $k$  independent delivery trucks of identical capacity  $C$ . Delivery is to be accomplished at the minimum total cost, with  $c_{ij} \geq 0$  denoting the transit cost from  $i$  to  $j$ , for  $0 \leq i, j \leq n$ . The cost structure is assumed symmetric, that is,  $c_{ij} = c_{ji}$  and  $c_{ii} = 0$ . Obviously, a solution for this problem consists of a partition of  $N$  into  $k$  routes  $\{R_1, \dots, R_k\}$ , each satisfying  $\sum_{j \in R_i} d_j \leq C$ . This problem is naturally associated with the complete undirected graph consisting of nodes  $N \cup \{0\}$ , edges  $E$ , and edge-traversal costs  $c_{ij}, \{i, j\} \in E$ . In this graph, a solution is the union of  $k$

cycles whose only intersection is the depot node. Each cycle corresponds to the route serviced by one of the  $k$  trucks. By associating a binary variable with each edge in the graph, the integer linear programming (ILP) formulation to minimize a total travel distance was obtained as follows (Ralphs *et al.*, 2003) :

$$\min z_1 = \sum_{e \in E} c_e x_e \quad (1)$$

s.t.

$$\sum_{e = \{0, j\} \in E} x_e = 2k \quad (2)$$

$$\sum_{e = \{i, j\} \in E} x_e = 2 \quad \forall i \in N \quad (3)$$

$$\sum_{e = \{i, j\} \in E, i \in S, j \notin S} x_e \geq 2b(S) \quad \forall S \subset N, |S| > 1 \quad (4)$$

$$x_e : \text{Binary} \quad \forall e = \{i, j\} \in E, i, j \neq 0, i \neq j$$

To bound Constraint (4),  $b(S) = \lceil (\sum_{i \in S} d_i) / C \rceil$  is

defined and it serves as an obvious lower bound on the number of trucks needed to serve the customers in  $S$ . Constraints (2) and (3) are the degree constraints. Note that constraints (2) and (3) can turn into an unfavorable instance resulting from an exceptional delivery situation where  $k = 1$  and  $N = 1$  that is, when one route for a customer must be scheduled, (2) and (3) turn out to be identical. Constraints (4) can be viewed as a generalization of the sub-tour elimination constraints from the TSP namely the capacity constraints and serve to enforce the connectivity of the solution, as well as to ensure that route has total demand not exceeding the capacity  $C$ . A stronger inequality may be obtained by computing the solution to a Bin-Packing Problem (BPP) (Even *et al.*, 2003) with the customer demands in set  $S$  being packed into bins of size  $C$ .

Moreover, we can minimize the difference between the maximum travel distance and the minimum travel distance. Suppose  $pathD(R_i)$  to denote the path length of  $R_i$ . Then,

$$\min z_2 = CB - CS \quad (5)$$

where,  $CB = \max\{pathD(R_i)\}$  and  $CS = \min\{pathD(R_i)\}$ ,  $i = 1, \dots, k$ . That is,  $CS$  and  $CB$  are the minimum daily distance and maximum distance respectively.

The objective function (5) enforces the gap between the maximum distance and minimum distance so as to make loads of truck balanced resulting in minimizing the variance among loads.

From (1) and (5), we can induce one multi-criteria objective function as follows :

$$\min z = w_1 z_1 + w_2 z_2 \quad (6)$$

where,  $w_1 + w_2 = 1$ ,  $w_1, w_2 \geq 0$ . The objective

function (6) is a weight sum of a pair of criteria where the weight  $w_1, w_2$  can be specified to weigh the contribution of each goal differently in the overall score.

It is obvious that the CVRP is closely related to two difficult combinatorial problems. By setting  $C = \infty$ , we get an instance of the Multiple Traveling Salesman Problem (MTSP). On the other hand, the question of whether there exists a feasible solution for a given instance of the CVRP is an instance of the BPP. The decision version of this problem is conceptually equivalent to a CVRP model in which all edge costs are taken to be zero. Hence, we can think of the first transformation as relaxing the underlying packing (BPP) structure and the second transformation as relaxing the underlying routing (TSP) structure. Because of the interplay between the two underlying models, instance of the CVRP can be difficult to solve in practice. As a result, the CVRP involving load balancing criteria can be much extremely difficult to solve in real situation.

To address the inherent intractable complexity of the CVRP with load balancing problem, we can take into account using the market based approaches. Auctioning process allows the load traders to bid for products based on the cost of adding the products to itself. Since the architecture is based on the concept of an intelligent agent, we now look at the agent technology prior to proceeding.

An agent is a software component and has many attributes associated with being intelligent (ACM, 1999; IEEE, 1996). The attributes are summarized in <Table 1>.

**Table 1.** Attributes of an intelligent agent

Attribute	Definition
Anticipate	Ability to anticipate changes in the environment and act accordingly
Communicative	Ability to communicate with other agents in a closed loop system
Autonomous	Ability to control over its own actions and to make decision on its own
Reactive	Ability to sense changes in the environment and act accordingly to those changes
Goal Driven	Each agent has a goal and is sometimes pro-active
Mobile	Ability to travel from one host to another without requiring human intervention
Learning	Ability to adapt in accordance with previous experience
Persistent	Capability to live for long periods of unattended operation
Collaborative	Ability to interact with humans and other agents
Distributed	Ability to survive under the environment free from any central authority governing the agent

Intelligent agents have been used to solve many kinds of problems such as : news finder, distributed component libraries, web search tools, intelligent routers, semantic broker and name-space services, dynamic cataloguing services, electronic commerce with message exchange, interface animation, tools with learning and knowledge sharing capabilities, robot control, project management and supply chain management (Huhns and Singh, 1997).

For applying the agent concept to CVRP with load balancing, we need an architecture that has to consist of a negotiation principle and a coordination framework to support it. The negotiation principle is the spirit of the agent-based models. It broadcasts the availability of the new products (i.e., customers) and accepts the bids on the products and commits the product to the bid with the lowest cost. Meanwhile, the coordination framework should be designed for decentralized systems of self-interested and rational entities (Thangiah *et al.*, 2003).

Market based automated negotiation, or more commonly *market based control*, is a paradigm for controlling complex system that would otherwise be very difficult to handle, by taking advantage of some desirable features of a market (especially a free market) including decentralization, interacting agents, and some notion of resources that need to be allocated (Clearwater, 1996). This approach has been applied to a wide range of fields such as supply chain management (Hinkkanen *et al.*, 1997; Sauter *et al.*, 1999), vehicle routing (Sandholm, 1993), manufacturing scheduling (Tilley, 1996), and process control (Jose and Ungar, 1998). The load assignment problem domain can be classified into the case of distributed resource allocation having a team objective.

In this paper, we consider two market based control mechanisms (Lee, 2002) that are designed as follows :

- Double Auction (DA) : Within the control window, multiple sellers and multiple buyers place or ask bids for the exchange of a designated commodity in a virtual market which matches buyers and sellers immediately on detection of compatible bids
- SpotMarket : A market is established for a seller and a buyer within the control window, meaning that the seller sells any commodity that is most benefit to the buyer.

In general, this paper improves on the previous result (Oh *et al.*, 2005) which showed that even blind market trading can be useful when information is typically less assured and changed in an unexpected way but did not show solid results to prove that their proposal can be comparable to pre-existing approaches in terms of various performance measures.

Based on this setting, we can define our load problem in detail. In general, when considering the load assignment problem, the following definitions hold :

**Definition 1.** Let **Load** denote the graph  $\langle V, E \rangle$  which contains  $V$  as a set of products intended to be delivered to their destinations and  $E$  as a set of distances between destinations,  $v_i (\in V)$  and  $v_j (\in V)$ .

**Definition 2.** Let  $TD(\text{Load}_i)$  denote the length of the shortest travel path of  $\text{Load}_i$   $i=1, \dots, k$ .

**Definition 3.** Let  $V(\text{Load}_i)$  denote the number of products deliver  $\text{Load}_i$ .

**Definition 4.** Let **LoadMakeup** denote a set of  $\text{Load}_i$   $i=1, \dots, k$ . such that if  $W$  products intended to be delivered from the yard to customers, then  $\sum_{i=1}^k V(\text{Load}_i) = W$ . When a **LoadMakeup** is obtained,  $\sum_{i=1}^k TD(\text{Load}_i)$  is the total distance required to deliver  $W$  products.

**Definition 5.** **Load Trader  $i$**  is an agent that possess a **Load  $i$**  and participates in **DA** or **SpotMarket** in order to minimize : (1)  $TD(\text{Load}_i)$  locally, or (2)  $\sum_{i=1}^k TD(\text{Load}_i)$  globally by cooperation with other **Load Traders**.

Using the definitions above, we aim at solving the following problems.

Given a set of products  $W$ , effectively find a **LoadMakeup** such that for  $\forall \text{Load}_i \in \text{LoadMakeup}$  (1)  $\sum_{i=1}^k TD(\text{Load}_i)$  is minimized; or (2) variance w.r.t  $TD(\text{Load}_i)$  is minimized.

In particular, the second problem of finding a **LoadMakeup** with minimized variance over **TD (Load)** is what we refer to as the **Load Balancing** problem, and can be easily formulated by balance spanning tree (BST). A spanning tree  $T$  of a graph  $G$  is called a BST if it minimizes the difference between the most costly arc and the least costly arc selected (that is, from among all spanning trees of  $G$ , the difference between the maximum arc cost in  $T$  and the minimum arc cost in  $T$  is as small as possible). Obviously, one very inefficient (if not intractable) method of finding a BST is to enumerate all spanning trees of  $G$  and then select the balanced spanning tree from among these. Since we aim at obtaining a set of **Loads** rather than discovering just a single balanced shortest path, such existing BST solution is, however, not feasible.

In the following section, we investigate alternative, non-exhaustive, but approximate solution based on aforementioned **DA** or **SpotMarket** using Clark-Wright method to generate initial endowment.

### 3. LoadMARKET Model

In handling our problems with market based control, two issues are critical : (1) the initial **Loads** to be assigned to **Load Traders** for their endowment ; and (2) the market mechanism to control **Load Traders**. To address the issues, we propose following techniques –generating initial load assignment using Clark-Wright (Bodin *et al.*, 1983) algorithm referred to as the savings method and two market based control algorithms. Note that from the graph theory perspective, trading a car between **Load Traders** can be viewed as a synchronized 2-opt Exchanges (Reinelt, 1994) because through sending their cars to partners, each trader need to remove two edges associated with node representing the exporting car and in turn, traders reconnect edges associated with the node corresponding to the importing car.

#### 3.1 Clark-Wright

The Clark-Wright algorithm initially starts with assuming that for a set of  $N$  customers,  $N$  vehicles are required. The main procedure of this method is to compute the savings  $s_{ij}$  which is obtained by merging two customers (e.g., cities)  $i$  and  $j$  so that one truck can serve both. Suppose that  $d_{ij}$  is the distance between  $i$  and  $j$ . Then, the savings  $s_{ij} = d_{0i} + d_{0j} - d_{ij}$  where  $d_{0i}, d_{0j}$  are distances from the depot to  $i$  and  $j$  respectively.  $s_{ij}$  implies that we can save one vehicle to serve  $i$  and  $j$  at once but at the expense of increasing the distance to be traveled by the vehicle in order to serve  $i$  and  $j$ . For each pair of  $i$  and  $j$ ,  $s_{ij}$  is calculated and the customers with highest savings are sorted out and merged as long as the capacity restrictions are not violated. This process continues until no more merging is available. The Clark-Wright method operates in a centralized way. Therefore, when it starts and ends, it is quite possible that the completely changed delivery schedules can emerge. Since this complete change can cause another overhead, we will look at other solutions that work on top of a solution generated by the Clark-Wright but operated in adaptive and distributed way. In the next Section, we suggest two market based algorithms that are designed to enhance the Clark-Wright initial solution. Note that following two algorithms are not competitive with the Clark-Wright algorithm but complementary for the algorithm. Nonetheless, we will show the performance enhancement caused by the cooperative activity between the Clark-Wright algorithm and the market based methods through experimental validation.

### 3.2 LOADMARKET<sup>DA</sup>

This algorithm uses an auctioning process to re-assign cars to each transportation trucks. The auctioning process mainly consists of :

- Task announcement : A market agent announce to each load agent on the availability of a product.
- Bid collection : Bids for the product are collected from load agents
- Bid evaluation : After bids are collected by the partner load agent, the cost vs. benefit are computed and evaluated.
- Task commitment : After selecting the most rewarding bid and its load agent, the product is exchanged or moved.

The main idea of this algorithm lies on the post process to run DA (i.e., double-sided market between Load Traders) to enhance an initial assignment generated by the Clark-Wright method such that each Load can make their TD (Load) shorter.

<p><b>Input :</b> <math>W</math> : all products, <math>E</math> : distance matrix, and <math>\#</math> : one load size</p> <p><b>Output :</b> <math>L</math> : a set of loads</p> <p><math>L \leftarrow \emptyset</math></p> <p><math>L' \leftarrow \text{LOADMARKET}^{\text{MST}}(W, E, \#)</math></p> <p><b>while</b> <math>\text{Converged} \neq \text{true}</math> <b>do</b></p> <p style="padding-left: 2em;"><math>L' \leftarrow \text{DA}(L')</math>;</p> <p><b>end</b></p> <p>print <math>L(\leftarrow L')</math></p>
---

#### Algorithm : LOADMARKET<sup>DA</sup>

As shown in <Figure 3>, when a new iteration of DA begins, a Virtual Market sends out “Do Sync” signals to all Load Traders. Next, since Load Traders have an option to choose one role from either a buyer or a seller, they pick up one role and respond to the Virtual market. Subsequently, the Virtual market matches buyers and sellers. For matched buyer and seller, they start a negotiation to decide on exchanging their products. If exchanging activity is beneficial to both of them, they immediately conduct the exchanging activity.

All traders have a common policy, that is, *minimize TD (Load)*. With this policy, the DA protocol exchanging process iterates until the market converges or a pre-defined iteration is satisfied. The iterating DA protocol process is expected to show emergent behavior that is, a more efficient load assignment (i.e., more reduced travel distance).

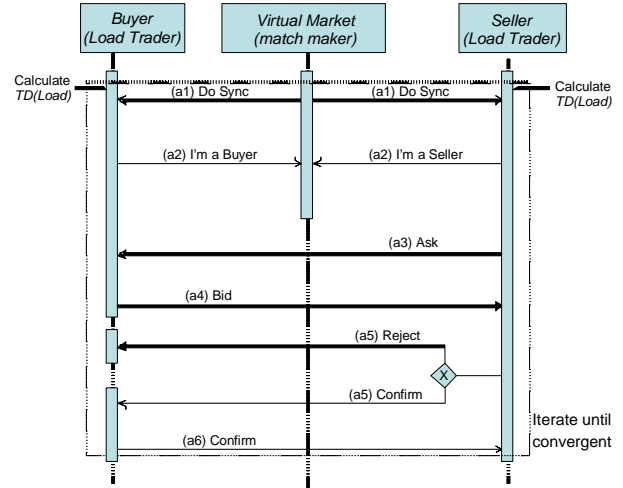


Figure 3. DA protocol.

### 3.3 LOADMARKET<sup>SpotMarket</sup>

Like LOADMARKET<sup>DA</sup>, it also applies a post process to the initial assignment created by the Clark-Wright method. One thing different from LOADMARKET<sup>DA</sup> is that it runs afore-defined SpotMarket mechanism which allows only two Load Traders to trade each other. In this context, one Load Trader with the most costly TD (Load) becomes a buyer while the other Load Trader with the least costly TD (Load) becomes a seller. The seller's role is focused on supporting the buyer by allowing the buyer to exchange any product which the buyer wants as long as her loss is not greater than some threshold. In other words, the seller's policy is to *minimize*  $\max_i TD(\text{Load}_i)$  such that load balancing on the corresponding LoadMakeup is obtained even though she can take some loss. The definition of LoadMakeup is found in the previous section.

<p><b>Input :</b> <math>W</math> : all products, <math>E</math> : distance matrix, and <math>\#</math> : one load size</p> <p><b>Output :</b> <math>L</math> : a set of loads</p> <p><math>L \leftarrow \emptyset</math></p> <p><math>L' \leftarrow \text{LOADMARKET}^{\text{MST}}(W, E, \#)</math></p> <p><b>while</b> <math>\text{Converged} \neq \text{true}</math> <b>do</b></p> <p style="padding-left: 2em;"><math>L' \leftarrow \text{Auction}(L')</math>;</p> <p><b>end</b></p> <p>print <math>L(\leftarrow L')</math></p>
--

#### Algorithm : LOADMARKET<sup>SpotMarket</sup>

Compared with LOADMARKET<sup>DA</sup> which invokes the Load Traders to seek their immediate increased benefit (i.e., greedy behavior), LOADMARKET<sup>SpotMarket</sup>



drives the Load Traders to cooperate each other. <Figure 4> shows the protocol between the buyer and seller. This SpotMarket protocol exchanging process iterates until the market converges or a pre-defined iteration is satisfied.

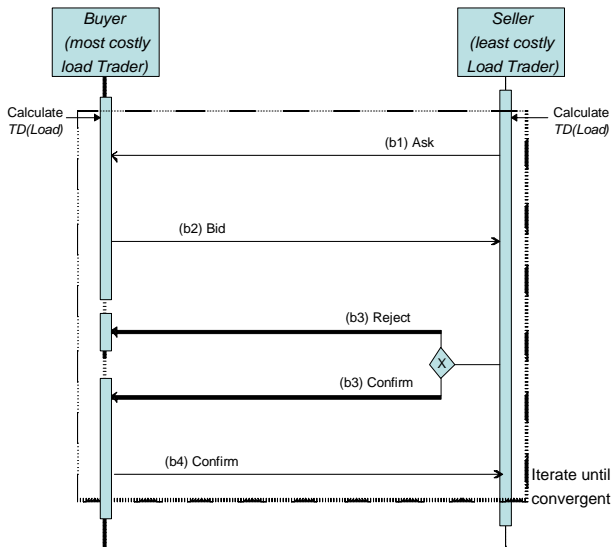


Figure 4. SpotMarket Protocol.

## 4. Experimental Validation

### 4.1 Methodology

In order to avoid the overheads caused by finding shortest paths during runtime for generating LoadMakeup, we made up with incidence matrices which specify the distance between cities earlier. The process of incidence matrix generation is as follows : First, we develop a random graph (Denning, 2004) consisting of nodes as cities and edge as roads between cities. Each edge has weight as 1 meaning that if starting point is  $i$  and end point is  $j$ , then the distance from  $i$  to  $j$  or from  $j$  to  $i$  becomes 1. Second, the incidence matrix are generated by computing the shortest path between each  $i$  and  $j$ . Thereby, the resultant incidence matrix enables LoadMarket to refer the shortest distance between cities  $i$  and  $j$  without additional searching time overhead. <Figure 5> illustrates this generation process. Note that we can vary the random graphs with number of cities and number of edges. For example, we can get more dense network if we increase the number of edges as shown in <Figure 5> where (50, 75) pair appears more packed than (50, 55) thereby guarantying the high chance of finding much shorter path between cities than otherwise. For the experiment purpose, we generated three cases such as : (50, 55), (100, 110) and

(200, 220). In the (50, 55) case, 50 cities and 55 roads are randomly linked between cities. Similarly, (100, 110) and (200, 220) can be defined.

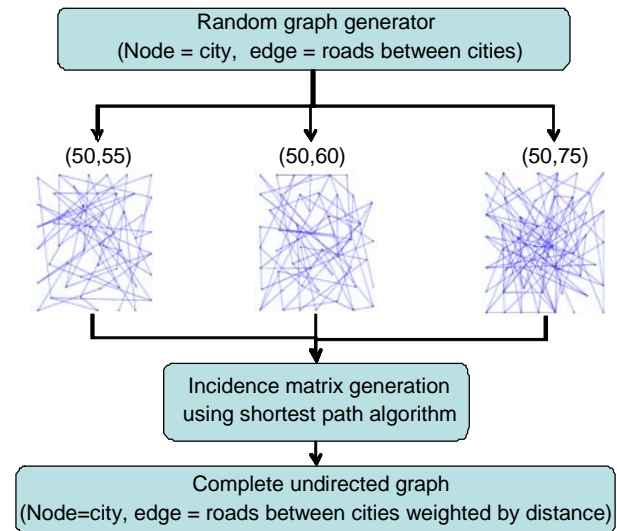


Figure 5. Incidence matrix generation for experimental purpose.

### 4.2 Set-up

To validate the efficacy of our proposals, we conducted experiments using simulation. First, we built a simulator that operates according to scenarios identical to the motivation example in the introduction Section. In other words, the simulator mainly consists of three parts : (1) a plant, (2) a yard and (3) a transportation system (i.e., a set of trucks). Every day, the plant produces a 50 limited number of cars that move into yard where cars waiting for their delivery. Each produced car has its delivery destination indexed by city  $i$  generated randomly. We assume that the daily available number of trucks is five so that a LoadMakeup can be generated daily such that ten cars are wrapped in one Load together and contained in one truck. Each truck carries ten cars and delivers them to destination cities.

We varied the number of cities as 50, 100, and 200 respectively while keeping the restriction on the daily production size as 50 cars. Note that we already built three symmetric incidence matrices which provide the shortest path distance between cities for each three different city number case. More detailed, in the case of 100 cities, each city is assigned with unique index between 1 and 100, and the distance between any two cities,  $i$  and  $j$  are specified in the corresponding incidence matrix generated in the previously. Be aware of that since we assume that city 1 is considered as the depot, each among 5 trucks departs city 1 and delivers 10 cars and must return to the depot.

Overall, we experimented a combination of 3 different numbers of cities 3 different algorithms = 9 combinations. Note that the initial solutions generated by the Clark-Wright method are underlying two market based algorithms but for the comparison purpose, we run the Clark-Wright algorithm separately too. Each experiment for each combination has five replications. That is, for each combination, we generate 18,250 data (i.e., 50 cars per a day 365 days) with 5 replications using the simulator. Assigning a city index to a produced car follows the uniform distribution which is specified by the number of cities.

Regarding market control parameters, we assume that the protocol exchange iteration per a trading is limited by 100 times. It means that the protocol exchange process is forced to stop once the exchange iteration exceeds 100 even though the process does not reach the convergence.

In this experiment, we assume the assurance of information is 100% and all truck departure time are identical because if we involve those factors, it is extremely non-trivial to compare our proposal to the Clark-Wright algorithm which must run based on the fully-informed situation.

### 4.3 Results

<Figure 6> and <Table 2> summarize the average travel distance time of LoadMakeup with respect to three algorithms over three different numbers of cities, where all algorithms' performance get worse reasonably as the number of cities increases. From the result, we can realize that  $\text{LOADMARKET}^{\text{DA}}$  is uniformly better than other algorithms. The Clark-Wright method performs worse than others. <Figure 7> and <Table 3> summarize the standard deviation (STD) with respect to the travel distance between loads within the same LoadMakeup, where all algorithms' performance get worse as the number of cities increases.

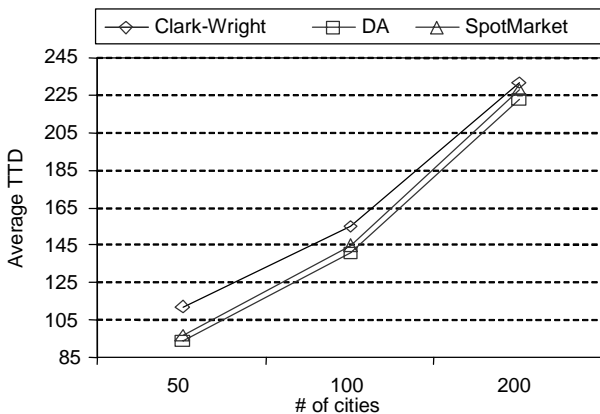


Figure 6. Average total travel distances.

Table 2. Average total travel distances

Algorithm	Number of cities		
	50	100	200
Clark-Wright	112.28	154.78	231.48
LOADMARKET <sup>DA</sup>	93.93	141.08	222.35
LOADMARKET <sup>SpotMarket</sup>	96.99	144.37	227.61

Interestingly, in this context,  $\text{LOADMARKET}^{\text{SpotMarket}}$  outperforms other algorithms uniformly. This is somewhat intuitive since  $\text{LOADMARKET}^{\text{DA}}$  stimulates Load Traders to act a greedy behavior seeking their immediate increased benefit while  $\text{LOADMARKET}^{\text{SpotMarket}}$  drives the least costly Load Trader to yield the most costly Load Trader its benefit. Similar to the total travel distance metric, the Clark-Wright turned worse in this measure.



Figure 7. STD w.r.t average total travel distances.

Table 3. STD w.r.t total travel distances

Algorithm	Number of cities		
	50	100	200
Clark-Wright	5.72	6.33	8.21
LOADMARKET <sup>DA</sup>	3.88	5.76	7.72
LOADMARKET <sup>SpotMarket</sup>	3.66	5.53	7.16

### 4.4 Implications and Lessons Learned

From the statistics and analysis above, we can look again at the applicability and relevance of the market based approaches to the intractable problem in practice. However, a general caveat is in order before we proceed to list the lessons learned from this results. That is, it is entirely possible that the result of experiment can be changed by the initial endowment for each load agent that is generated by an algorithm responsible for dividing products into number of loads.



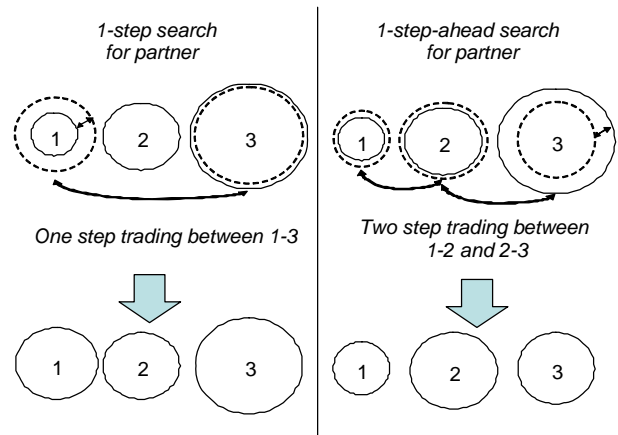
For example, if we had used meta-heuristics such as genetic algorithm to enhance the initial solutions provided by Clark-Wright algorithm, we could have achieved more enhanced final results. In addition, we could have applied more sophisticated bargaining algorithm than simple blind market based controls used in this paper. Nevertheless, we believe that it is worthwhile to evaluate our simple proposals in CVRP with load balancing from the perspective of discovering a new potential application for the market based control.

**Total travel distance** : As expected, we found that  $LOADMARKET^{DA}$  is most competitive as far as the total distance metric is concerned. Since the Clark-Wright is a kind of greedy algorithm, it could fall into a skewed solution. However, we overcome the skewed solution problem by applying the double auction mechanism that encourages each load trader to obtain a rewarding product through exchanging their less rewarding products with other load traders. In this architecture, the amount of value enhancement can be closely in proportion to the total number of negotiations. Because  $LOADMARKET^{DA}$  allows the load traders to continue the double auction process until a convergence is reached, the number of exchange must be larger than  $LOADMARKET^{SpotMarket}$ . However, this allowance could be a critical shortcoming in the sense that many real practices require real time based response from their load assignment scheduler. That is why we limit the iteration time as short as 100.

**Load balancing** : One somewhat surprising results of our analysis is that  $LOADMARKET^{SpotMarket}$  showed most competitive performance in terms of the load balancing metric despite of its simpler mechanism and faster computation than  $LOADMARKET^{DA}$ . It could be useful information to people who have to solve CVRP with load balancing in fast time or real time base.

**Together** : According to our results, two goals (i.e., the total distance of traveler and the standard deviation between loads) were not able to be achieved by one algorithm. In fact, there are many research opportunities how to trade-off two different objectives by controlling agents that are inherently driven in a greedy way to decrease their own travel distance at the expense of increasing travel distance of other agents. For example, we can add some intelligent coordination ability to each agent and put another agent dedicated to coordinate tasks. It is true that our proposal is using 1-step search for an agent to look for its trading partner. In the worst case, some unfavorable situation happens as shown in <Figure 8> (left) where trading limited by 1-step search generated worse solution than the prior solution. However, if an intelligent agent can compute a two-step trading plan as shown in <Figure

8> (right) such that first trading is done between 2 and 3 and subsequently second trading is done between 1 and 2, it can make much better solution than the simple 1-step search case. However, the increased search steps may compensate computational resource exponentially. Nonetheless, the broader the search space for alternative partners is, the earlier solutions come out.



**Figure 8.** The different results caused by different search ways to find out trading partners.

## 5. Conclusions

In this paper, we have considered the problem of finding an efficient load assignment, and presented our proposals using the market based controls such as DA and SpotMarket and compared our proposal with Clark-Wright’s savings algorithm. One interesting observation is that despite its less competitive performance of the average travel distance,  $LOADMARKET^{SpotMarket}$  algorithm showed the best performance as far as the balanced load assignment is concerned.

Several directions are ahead for further research. Rather than the one-day based assignment we dealt with in this paper, the real time based dynamic load assignment can be considered next. For example, whenever a product is produced from the plant, our proposals will be applied and the load assignment can be updated continuously. In that context, less assured information and unexpected events can be involved.

Also, we can formulate and address another problem to find an optimal iteration number of protocol exchange per trading.

## Acknowledgements

This research has been done with support from General Motors Research and Development Center.

## References

- ACM (1999), Multi-agent systems on the Net and Agents in E-commerce, *Communications of the ACM*, **42**(3).
- Bodin, L., Golden, B., Assa, A., and Ball, M. (1983), The state of the art in the routing and scheduling of vehicles and crews, *Computers and Operations Research*, **10**(2), 63-211.
- Chung, E. Y. and Park, Y. B. (2004), A Genetic algorithm for vehicle routing problems with mixed delivery and pick-up, *Journal of Korean Institute of Industrial Engineers*, **30**, 346-354.
- Clark, G. and Wright, J. W. (1964), Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, **12**, 568-581.
- Clearwater, S. H., Costanza, R., Dixon, M., and Schroeder, B. (1996), Saving energy using market-based control, In *Market-Based Control - A Paradigm for Distributed Resource Allocation*, World Scientific, Singapore.
- Denning J. P. (2004), Network Laws, *Communications of the ACM*, **47**(11).
- Even, G., Garg N., Konemann, J., Ravi, R., and Sinha, A. (2003), Covering graphs using trees and stars, In *Proceedings of RANDOM-APPROX*.
- Flood, M. M. (1956), The Traveling Salesman Problem, *Operations Research*, **4**, 61-75.
- Hinkkanen, A., Kalakota, R., Saengcharoenrat, P., Stallaert, J., and Whinston, A. B. (1997), Distributed decision support systems for real-time supply chain management using agent technologies, *Readings in Electronic Commerce*, AW Computer and Engineering Publishing Group.
- Huhns, M. N. and Singh, M. P. (1997), *Readings in agents*, Morgan Kaufman.
- IEEE (1996), Intelligent systems and their applications, *IEEE Expert* December issue.
- Jose, R. A. and Ungar, L. H. (1998), Auction-driven coordination for plant wide optimization, In *Proceedings of Foundations of Computer-Aided Process Operation (FOCAPO)*, Snowbird, UT, USA.
- Lee, Y-H. (2002), Market-based dynamic resource control of distributed multiple projects, *Ph.D. thesis, Department of Industrial Engineering, The Pennsylvania State University*, PA, USA.
- Oh, S-C., Yee, S-T., Kumara, S., and Tew, J., A family of market-based shipment methodologies for delivery supply chain, In *Proceedings of the 2005 Winter Simulation Conference*, USA.
- Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter L. E. (2003), On the capacitated vehicle routing problem, *Mathematical Programming Series B* 94.
- Reinelt, G. (1994), The traveling salesman, *Computational Solutions for TSP Applications*, Lecture Notes in Computer Science, 840, Springer-Verlag, Berlin Heidelberg New York.
- Ro, I. K. and Ye, S. Y. (1996), A Heuristic for vehicle routing problem, *Journal of Korean Institute of Industrial Engineers*, **22**, 325-336.
- Sandholm, T. (1993), An implementation of Contract Net Protocol based on marginal cost calculations, In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington DC, USA.
- Sauter, J. A., Parunak, H., Van, D., and Goic, J. (1999), ANTS in the supply chain, In *Workshop on Agent for Electronic Commerce at Agents '99*, Seattle, WA, USA.
- Thangiah, S. R., Shmygelska, O., and Mennell, W. (2003), An agent architecture for vehicle routing problems, In *Proceedings of ACM Symposium on Applied Computing*, Las Vegas, USA.
- Tilley, K. J. (1996), Machining task allocation in discrete manufacturing systems, In *Market-Based Control - A Paradigm for Distributed Resource Allocation*, World Scientific, Singapore.
- Yee, S-T. (2002), Simulation applications in the automotive industry : establishment of product offering and production leveling principles via supply chain simulation under order-to-delivery environment, In *Proceedings of the 2002 Winter Simulation Conference*, USA.
- Yoo, Y. S. and Ro, I. K. (1999), Vehicle routing problem using parallel genetic algorithm. *Journal of Korean Institute of Industrial Engineers*, **25**, 490-499.