Improving Quality of Crane-Lorry Assignments With Constraint Programming

Andy Hon Wai Chun, *Member, IEEE*, and Rebecca Y. M. Wong

***Abstract*—The ability to maximize service quality while minimizing cost is very important to service-oriented businesses, such as lorry leasing. Very often, the ability to consistently offer higher quality service is the main differentiating factor between a business and its competitors. For lorry leasing businesses, service quality and cost are directly related to how resources—vehicles, cranes, and drivers—are allocated to jobs. The ability to assign the right combination of resources is crucial to daily operations. This paper presents how we modeled this assignment problem as a constraint-satisfaction problem (CSP) and implemented using constraint programming (CP) with an algorithm that we call the crane-lorry optimizing engine (CLOE). CLOE was implemented for the largest crane-lorry company in Hong Kong. Furthermore, plans are generated within seconds compared to close to an hour if done manually. All necessary constraints and criteria are considered systematically. We have experimented with many different types of search heuristics and have analyzed their effects on plan quality. We have found that by considering both the experience of the crane-lorry-driver combination and maximizing on the order assignment, we were able to generate plans that were significantly better than those produced by human planners, and within a substantially shorter time. Although the use of constraint-based assignment techniques is still limited in the vehicle leasing industry, we hope our combination of automated assignment with Internet portal technology that streamlines business-to-business, business-to-consumer, and business-to-employee communications can excite interest in this area.**

***Index Terms*—Constraint programming (CP), resource allocation, scheduling, vehicle assignment.**

I. Introduction

A CRANE-LORRY is a combination of a lorry and a crane, the crane being usually permanently mounted on that lorry. The number of possible combinations of crane to lorry can be quite large and varied. The exact vehicle to be dispatched to service at customer's request depends on many factors, including the workload requirements, such as the size and weight of the payload, the storage location of the payload, etc.

The types of cranes used on a crane-lorry may also vary from company to company. Cranes are categorized in terms of ton·meters, which is the maximum capacity multiplied by the maximum outreach. For example, a crane that can lift 8 tons with an outreach of 2 m is rated at 16 ton·m. Cranes may range from 1 to 140 ton·m or even larger. In Hong Kong, the industry standard is to measure cranes in terms of an outreach of 2 m. For example, a 140 ton·m crane can maximally lift 70 tons of goods. Besides lifting and outreach capabilities, cranes are also classified according to its mechanical structure, such as knuckle crane, pick-up crane, hydraulic crane, etc. The specific needs of each job must be carefully considered to ensure that the right crane with the right set of capabilities is dispatched.

Since many different types of cranes are involved, each with different features and operational procedures that might be substantially different from each other, crane operators must be adequately trained on and licensed for operating that particular equipment before they can be assigned to drive that crane-lorry. To minimize the potential for operational errors, crane-lorry companies usually fix the assignment of a crane operator to a particular lorry for a period of time. The assignment plan is changed only occasionally when there is a new crane-lorry or staff, when staff become unavailable, or when the equipment breaks down. Since the driver assignment plan changes on an *ad hoc* basis, this plan is not generated by our system. Instead, it assumes driver assignment to be a manually maintained table in the system.

Lorries are classified according to their physical dimensions and their maximum gross vehicle weight (GVW), which is simply the empty vehicle and crane weight, plus the weight of the payload and crew. A 24-ton lorry may have ten wheels, while a 30-ton lorry may have 12 wheels. Different cranes have different requirements on the type of lorry that it can be mounted on. The dimension and weight of the payload will affect which vehicle should be assigned to service that job.

Cranes with small lifting capability, say 16 ton·m cranes for example, can be installed on most lorries provided that the lorry meets the minimum mounting specification of the crane. But cranes with larger lifting capability, such as 110-ton·m cranes, are usually only installed on larger crane-lorries, due to minimum axle requirements to keep the lorry stable during crane operations.

Cranes can be installed either at the front (behind the cab), center, or rear of a lorry. Crane-lorries with a crane mounted in the front can carry larger or longer goods, but are usually considered as less flexible or less versatile; the cab of the lorry blocks part of the crane's path of movement. On the other hand, crane-lorries with the crane mounted at the rear are considered more versatile as crane movement sweeps a wider area. However, they are limited by the size and length of the goods that they can carry. Reserving more versatile vehicles to handle more difficult jobs is one factor that is often considered by human planners in vehicle assignment.

Manuscript received June 22, 2003; revised December 10, 2004. This work was supported in part by the Research Grants Council of Hong Kong SAR, China under Project 9040517, CityU 1109/00E and in part by the City University of Hong Kong under Project 7001286. This paper was recommendad by Associate Editor C. White.

A. H. W. Chun is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (e-mail: andy.chun@ ieee.org).

R. Y. M. Wong is with the Sun Hung Kai Financial Group, 88 Queensway, Hong Kong, China (e-mail: rebeccawong33@gmail.com).

Digital Object Identifier 10.1109/TSMCC.2006.887013

1094-6977/$25.00 © 2007 IEEE

The assignment of lorries to customer orders is the most critical daily task for any crane-lorry leasing company. The plan must first of all be accurate—assigning the "right" vehicle-equipment combination to satisfy each customer order. It must also be optimizing—assigning the crane-lorry with just the right level of capabilities, not too much and not too little. Otherwise, some jobs might need to be subcontracted out due to lack of suitably equipped vehicles. This not only means lost revenue, but even worse, lost customers who might not return. Besides considering the features and capabilities of the lorry and the mounted crane, the human planner also needs to consider the priority of the customer, the past order history, geographical location, equipment preferences, and any other special requests.

The assignment application, described in this paper, was designed both as an infrastructure to streamline operations, through the "business to business to customer to employee" (B2B2C2E) exchange, and as an optimization engine to maximize utilization and customer satisfaction.

II. System Architecture

The system architecture of our crane-lorry scheduling system (CLSS) is quite similar to many other Internet exchanges. It has a multitiered architecture with Web servers, application servers, and database servers, all behind firewalls.

Only some of the features are accessible through the Internet, such as the B2C features for customer order submission. An "order," in our case, is a request for a crane-lorry with all the associated requirements, such as payload size and weight, type of crane needed, location, and date and time. Other more critical features are only accessible through an intranet by employees, such as generating an assignment plan and dispatching vehicles. Details of Sales Module, Scheduling Module, Subcontracting Module, and Dispatching Module will be described in Section III.

The exchange is somewhat different from typical B2C portals, as the system was designed to streamline the "whole" operation of a crane-lorry company. Besides streamlining communication between company and customer (B2C), the exchange also streamlines communication between company and its subcontractors (B2B) as well as between company and employees (B2E). We call this model a "B2B2C2E" exchange model.

In B2C mode, the "consumers"—customers and internal sales—can enter orders, modify orders, track order statuses, and process bills through the "Sales Module" provided by our system. (The internal sales handle orders that come through traditional channels of phone or fax.) For the "business" side of B2C, i.e., the operations managers, they can review customer orders, initiate planning, and produce management reports through the "Scheduling Module." These two modules will be described later in detail in Section III.

In B2B mode, the first "B" is the crane-lorry company that operates the exchange. The B2B mode is used to communicate with subcontractors—the other "B." The "Scheduling Module" identifies which customer orders cannot be fulfilled by the crane-lorry company and must be subcontracted. Subcontracting is a common practice in Hong Kong, and is performed daily in the company that we performed this research on. Subcontractors use the "Subcontracting Module" to receive job information, accept/reject jobs, and to enter job statuses, which can then be retrieved by the customers of the company that subcontracted the job, i.e., in the B2C mode.

Finally, the B2E mode is used to streamline communication between the operations manager (the "business") and the vehicle driver (the "employee"). The "Dispatching Module" handles the job dispatching, which can be over wireless devices, such as wireless application protocol (WAP) phones, two-way pagers, or PDAs.

III. Software Architecture

As mentioned previously, there are four key software modules in the system—the Sales Module, Scheduling Module, Subcontracting Module, and the Dispatching Module.

1) *Sales Module:* The "Sales Module" is responsible for maintaining customer and order information. It receives customer orders, handles customer queries, and provides order tracking. It is used either directly by the customer through the B2C portal or operated by an internal sales staff. The B2C portal is mainly used by regular customers who are already familiar with the different types of cranes and lorries that are available. Casual customers will normally call in and ask for assistance. The sales staff provides product information, such as which types of lorries and cranes are available, and their capabilities and differences. After an order is finalized, it is passed to the "Scheduling Module," which automatically assigns a suitable crane-to-lorry combination to fill that order.
2) *Scheduling Module:* The "Scheduling Module" is responsible for resource allocation using our crane-lorry optimizing engine (CLOE) assignment algorithm. Although the assignment algorithm was designed to operate in both dynamic real time and planning modes, it is only used in planning mode at this crane-lorry company. In practice, vehicle assignments are not made on the spot, but batched together into a daily pool of requests. The assignment algorithm then performs crane-lorry assignment at a predefined time each day. It will allocate lorries to customer orders according to customer needs, driver experience, and other business and operational rules and considerations. In addition, the "Scheduling Module" will decide whether subcontractors are needed and which orders are to be contracted out. Subcontractors are usually other smaller crane-lorry companies or individual lorry operators. The results of the assignment process are: 1) which lorries are still available; 2) which orders need to be subcontracted out; and 3) which lorry is assigned to which order. The assignment results will then be passed to the Sales Module, Subcontracting Module, and Dispatching Module, respectively, for further processing.
3) *Dispatching Module:* The "Dispatching Module" is responsible for dispatching orders to drivers and for receiving confirmation from the drivers on job statuses. Dispatching is done through mobile wireless devices. The

Dispatching Module dispatches jobs to drivers in the morning after the Scheduling Module finishes the vehicle assignment.

4) *Subcontracting Module:* The "Subcontracting Module" subcontracts excess jobs or jobs it cannot handle out to other companies. From the Scheduling Module, it receives a daily list of orders that cannot be fulfilled with resources within the company itself. Most often, the company will subcontract jobs out to a predefined pool of "friendly" subcontractors that already have a business relationship with the company. The prices for different types of services are already preagreed upon in a long-term contract. All the subcontractors need to do is simply to confirm whether they have the capacity to handle the excess orders that have been passed to them. If not, the order gets sent to another subcontractor in the pool. The key principle behind subcontracting is of course to minimize revenue loss while maximizing quality of service. This means keeping the more important or "larger" jobs within the company itself.

In this paper, we will focus mainly on the "Scheduling Module," as that is where the constraint-satisfaction problem (CSP)-based CLOE assignment algorithm resides.

IV. Business Rules and Constraints

The business logic and business objectives will impact how the assignment algorithm should be designed. A crane-lorry company is basically a provider of transportation services. In the company that we performed this research on, each and every lorry has a permanently mounted crane. Depending on the crane, it can be used to lift different types of heavy payloads, such as construction material or even shipping containers. The payload is then transported from one site to another and usually within the same day or overnight, as all jobs are local. This company has around 30 crane-lorries, in different sizes and combinations. During the assignment process, a human planner will not only need to consider basic requirements, such as technical specifications of the crane and lorry, but also how to maximize both: 1) customer satisfaction and 2) resource utilization.

A. Customer Satisfaction

The measurement of customer satisfaction might be defined differently for different companies. Our CLOE algorithm is open and allows different measures to be dynamically defined to suit the needs of different companies. For this company, it is defined as a combination of customer preference and driver experience. Customers may have a special preference on a particular crane-lorry that it has used before. Assigning a driver with more experience can also increase customer satisfaction. Driver experience is defined as the number of times a driver-vehicle combination has performed a particular type of job before. This is computed using historical statistics on past job assignments. Having more experience means the job is more likely to be completed on time. The net effect this will have on the CLOE assignment algorithm is that it will provide some regularity and consistency in vehicle-driver assignments over time, which this company has found to be highly desirable among its customers.

B. Resource Utilization

Maximizing resource utilization means resources are not over or under provided. It also means that the company should try to handle as many of the orders itself and minimize the need for subcontractors.

C. Rules and Objectives

Based on the results of our business analysis, we identified a core set of business rules and objectives that are considered during crane-lorry assignment. The term "rule" is used loosely here. In actuality, a "rule" might be implemented as a constraint in a CSP model or as a search heuristic. Implementation details will be described in following sections of this paper. The basic business rules and objectives are as follows.

- *No overlap constraint*: Each lorry should only be assigned to one order at a time.
- *Fulfill workload constraint:* Lorry assigned to job must be adequately equipped for the job.
- *Maximize experience:* Most experienced lorry-driver should be assigned, whenever possible.
- *Maximize utilization:* Minimize number of orders given to subcontractors.
- *Minimize subcontract cost:* When subcontracting, use crane-lorry that minimally satisfies customer needs to reduce cost (considered by the "Subcontracting Module").
- *Minimize subcontract order size:* When subcontracting, subcontract orders that require smaller sized lorry to reduce cost.

V. Manual Assignment Process

Most of the orders will have arrived by the end of each business day. At that time, the human planner will begin to construct the plan for the next day. The planner will need to complete the assignments as soon as possible in order to determine which orders need to be subcontracted, and then to contact those companies for confirmation. Finding suitable subcontractors that have just the right equipment available and confirming the subcontract orders is a very time consuming process. The sooner a plan is produced, the better the chance of finding a suitable subcontractor. As time goes by, it will be more difficult to find suitable lorries, as other companies will have contracted them already. This implies that lorries will have to be leased from a more expensive company or a lesser quality, or larger lorry must be used to satisfy customer needs.

When the assignment process is performed manually, there is always a chance that mistakes might be made because a wide variety of constraints and criteria must all be considered at the same time and under tight time pressure. Mistakes might be over or under utilization of crane-lorries or not assigning a driver that the customer has worked with before and was happy with his/her performance. One of our key objectives in the design of the "Scheduling Module" is to reduce the assignment time and to improve the quality of resulting plans. Our CLOE algorithm produces assignments within seconds, while human planners take close to an hour to complete. Since all relevant constraints are considered, the "Scheduling Module" ensures that no mistakes can ever be made. Furthermore, there is no way an average human planner can thoroughly consider driver experience or historical data when performing assignment. There are only one or two "expert" planners who have been around long enough to remember enough details to produce high-quality assignments.

The measurement of how "good" an assignment plan is may differ from company to company. Again, our design is open to allow companies to dynamically adjust their "goodness" definition without any source code modification. "Goodness" for this company is determined by two key factors—whether the plan can: 1) minimize cost of outsourcing/subcontracting and at the same time 2) maximize quality of assignment/customer satisfaction.

The following sections details our CLOE algorithm that is part of the "Scheduling Module" that meets these design objectives and considers all the necessary business rules and constraints.

VI. Related Research

There are many different areas of optimization and scheduling-related research on transportation problems, such as vehicle routing problem (VRP), vehicle scheduling problem (VSP), automated guided vehicles (AGV), and vehicle assignment problem (VAP). These problems have played an important role in operations research (OR) ever since the works of Dantzig and Ramser [1] and Clarke and Wright [2] have been published. Our CLOE research falls within the VAP research area. Before we explain VAP, the following highlights other vehicle-related scheduling research to help delimit our problem space.

VRP is basically a type of traveling salesman problem (TSP) where we need to assign vehicles, instead of salesmen, to visit a set of customers to deliver and/or pickup goods. There are constraints on the vehicles, customers, drivers, goods, time, etc. The objective is to minimize cost, which is usually measured in terms of the number of vehicles and/or distance traveled and time spent. This classic problem has been solved with wider variety of techniques, including linear programming, constraint programming (CP) [3], genetic algorithms [4], artificial neural networks [5], heuristic approaches, etc.

For example, Shaw [6] used CP techniques combined with local search, in a way similar to shuffling technique [7], [8] used in job-shop scheduling to solve VRP. Ochi *et al.* [9] on the other hand, used parallel genetic algorithms combined with scattered search and a decomposition-into-petal procedure to solve a similar problem. Interestingly, Barnier and Brisset [10] offer another approach to solving VRP by combining both constraint-satisfaction techniques with genetic algorithm. This combined approach was shown to perform well even for large search spaces. Constraint satisfaction is used to compute feasible solutions in a subspace of the search space, while genetic algorithm is used on the space formed by these subspaces and to perform the optimization.

There are also different variations to VRP. For example, Frizzell and Giffin [11] use a heuristic approach to solve an extended VRP with split delivery, allowing delivery to a customer to be split between two or more vehicles, and a time frame when delivery must be made. Shang and Cuff [12] also used a heuristic algorithm to solve pickup and delivery problems within a Health Maintenance Organization (HMO). In this problem, vehicles are used to transport patient records, equipment, and supplies. The objective was to minimize vehicle expense, tardiness, and travel time. Their research showed that their heuristic algorithm performed better at meeting these objectives than the manual approach.

Since a wide variety of approaches have been tested on VRP, formal systems are being developed to help formalize and compare these approaches. For example, Desrochers *et al.* [13] outlined a formal system that can be used to define models and algorithms for vehicle routing and scheduling.

Related to the VRP is the VSP [14]. In simple terms, VSP is the problem of assigning vehicles to timetabled trips such that each trip is performed by one vehicle, a set of constraints is satisfied, and a cost function is minimized. The objectives usually include minimizing the number of vehicles and different operational costs, such as gas, driver, deadheading trips, etc. For example, bus scheduling is a typical VSP. Many real-life factors must be considered in solving VSP, such as the number of depots, travel time at different locations and times of the day, labor regulations that govern work hours of drivers, the types of vehicles available, refueling needs, etc. For example, Park and Song [15], and Park [16] used heuristic algorithms combined with a simple estimation model for time-varying travel times to solve VSP.

Closely related to VSP is driver scheduling, as drivers may have different skill sets, certifications, and experiences with handling different vehicles and equipment. Campbell and Diaby [17] outlined an assignment heuristic, based on linear assignment approximation, which provides one approach to the "beginning of shift" allocation problem for cross-trained workers. In our crane-lorry case, since safety in operating the crane equipment is utmost important, most of the driver assignments are fairly stable and need not be dynamically allocated each day.

A different set of problems related with vehicles and their scheduling is the problem of AGV. AGV are used in many different environments such as factories, container terminals, warehouses, etc, to transport goods and material from one location to another. For example, Shawik [18] described research work in flexible manufacturing system (FMS) and its related AGV scheduling problem using a heuristic approach. Ulusoy *et al.* [19] solves a similar problem for scheduling AGV within a FMS, by using genetic algorithms. Both Shawik and Ulusoy *et al.*, combined manufacturing machine scheduling with AGV scheduling. Other researchers focused just on AGV scheduling. For example, Wu [20] worked on a restricted AGV scheduling problem using problem decomposition combined with local search. Huang and Hallam [21] and Kwa [22], [23] used a totally different approach of distributed AI (DAI) and negotiation among software agents to perform conflict resolution within the AGV scheduling problem.

The work we performed in crane-lorry assignment is related to these research areas, but has some differences. We call this the VAP. It is a specialized combination of the VRP and VSP. In our lorry assignment problem, we need to satisfy pickup and delivery jobs by assigning vehicle-driver pairs to them. Unlike assumptions made by many VRPs and VSPs, our vehicles are not homogeneous and may contain different types of equipment. The routing is simplified, as vehicles are highly specialized and unique, and can only perform certain types of jobs. Hence, vehicles usually perform only one or two jobs a day. Unlike other vehicle-related scheduling research, such as VRP, VSP, or AGV that are more interested in optimality, using genetic algorithms for example, the key problem to be solved in VAP is not to compute the most efficient route or schedule, but to find the "most appropriate" assignment of equipment to different types of jobs while maximizing utilization and customer satisfaction. The "quality" of the assignment plays an important role as this can directly impact customer satisfaction. Hence, VAP can be treated as a type of resource-assignment problem. Most resource-assignment problems are modeled as CSPs and solved using CP, which is the case for CLOE.

Gill [24] in his Sunray V system, solved a related VAP for freight-forwarders also using CP techniques. Sunray V's scheduling algorithm determines how many jobs the company can handle, which jobs to subcontract, and assigns lorries or towheads to jobs. The key objective is to minimize subcontracted jobs, meet all job deadlines, and to minimize inter-job idle time. However, in the case of Sunray V, the towheads are pretty much homogeneous, unlike our crane-lorries. On the other hand, we need not schedule the jobs, as the customer dictates when the lorry is needed. For our problem, customer satisfaction and compatibility of driver's experience with job assignment is more important than minimizing travel time.

Another related research is that of Duncan's Schedule-IT [25] application. Schedule-IT also uses a constraint-based algorithm to allocate drivers and vehicles for the delivery of customer orders. Like our algorithm and that of Sunray V's, excess work will be subcontracted out to other companies. There are a number of constraints concerning the capabilities of the resources (vehicles and drivers) and limitation on working hours. Schedule-IT uses a driver-driven algorithm that schedules all the jobs of one driver before moving on to the next. In this way, the scheduler partitions the problem into subproblems to be solved. For our case, all the orders are solved in one global set. To meet our objective of maximizing the degree of match between driver experience and work order, our global approach will yield better solutions.

VII. Model and Implementation

CLOE is implemented using Java and a Java CP engine called JSolver [26]. JSolver extends the object-oriented programming paradigm of Java with constraint-based declarative programming. It allows the development of assignment and resource management systems in Java, and deployment of these systems over the Web or within an enterprise Java bean (EJB) application server. JSolver has been successfully used to solve different types of engineering problems as well as e-trade negotiation problems [27].

We have modeled the crane-lorry assignment problem as a CSP [28]–[32]. In general, any assignment or resource allocation problems can be formulated as a CSP, which involves the assignment of values to variables, subjected to a set of constraints. We have successfully used CSP algorithms for other transportation-related resource allocation problems before [33]–[35].

CSP can be defined as consisting of a finite set of n variables $v_1, v_2, \ldots, v_n$, a set of domains $d_1, d_2, \ldots, d_n$, and a set of constraint relations $c_1, c_2, \ldots, c_m$. Each d_i defines a finite set of values (or solutions) that variable v_i may be assigned. A constraint c_j specifies the consistent or inconsistent choices among variables, and is defined as a subset of the Cartesian product: $c_j \subseteq d_1 \times d_2 \times \cdots \times d_n$. The goal of a CSP algorithm is to find one tuple from $d_1 \times d_2 \times \cdots \times d_n$ such that n assignments of values to variables satisfy all constraints simultaneously.

When the lorry-assignment problem is formulated as a CSP, each variable or unknown represents the assignment of a lorry to an order. The domain of each variable, i.e., the possible orders a lorry may be assigned to, will initially contain the set of all orders for the next day. The CSP constraints are restrictions on how these lorries can be assigned based on business and operational rules and constraints.

A. CLOE Algorithm

The CLOE algorithm is a recursive search algorithm. At each recursion level, CLOE selects the next "best unassigned crane-lorry," and either assigns the next "best order to assign" or mark it as "unassignable." A crane-lorry is "unassignable" if there are no orders that it can fulfil. This is because either it is underequipped or its equipments do not fit the jobs for that day. This is different from being overequipped. Assignments can still be made if the lorry is overequipped. However, overequipped lorry should only be assigned if there are no other choices left, as the equipment will be underutilized or, even worse, make another order that really needs this resource to be unfulfillable.

Heuristics define what "best unassigned crane-lorry" means, as well as what "best order to assign" means. These heuristics guide the search and improve the quality of the resulting assignment plan. We further use a variant of the AC-3 arc consistency algorithm [29], [30] to perform constraint propagation and domain reduction to further improve search performance.

The following is the pseudocode for the CLOE algorithm using notations similar to those used by Tsang [28]. (The pseudocodes in this paper were designed purely for illustration purposes to highlight the key structure of the algorithms and might not necessarily be the actual or most efficient implementation used.) The notations are as follows:

- Z finite set of variables (set of all unassigned crane-lorries);
- D finite set of all variable domains (domain contains all the orders for the day);
- C finite set of constraints (restrictions on how lorries can be assigned).

```
PROCEDURE CLOE(Z, D, C)
BEGIN
  solution←NIL;
  D'←Propagate(Z, D, C, 0);
  IF (no domain in D' is empty)
  THEN solution←Search(Z, 0, D', C);
  RETURN (solution);
END
```

Fig. 1. CLOE procedure.

```
PROCEDURE Search(U, CL, D, C)
BEGIN
  fail←FALSE;
  IF(U={}) THEN RETURN (CL)   // all lorries have been assigned orders
  ELSE BEGIN
    x←SelectLorry(U);          // find next "best" lorry to schedule
    REPEAT
      v←SelectOrder(Dx);       // find next "best" order to assign
      Dx←Delete(v, Dx);        // delete selected order from Dx
      IF (CL+{<x,v>} violates no constraints)
      THEN BEGIN
        D'←Propagate(U-{x}, D, C, <x,v>);  // constraint propagation
        IF (no domain in D' is empty)
        THEN BEGIN             // recursively search
          result←Search(U-{x}, CL+{<x,v>}, D', C);
          IF (result≠NIL) THEN RETURN (result);
        END
        ELSE fail←TRUE;        // a domain in D' is empty
      END
      ELSE fail←TRUE;          // new compound label violates constraints
    UNTIL (Dx={} | fail=TRUE);
    RETURN (NIL);              // no solution
  END
END
```

Fig. 2. Search procedure.

The CLOE procedure (Fig. 1) first calls the *Propagate* procedure to perform domain reduction on posted constraints. This presearch arc-consistency check will improve performance by removing all "obvious" infeasible choices from the domains of the variables. Infeasible choices are defined through constraints contained in the set of constraints "C" that governs how lorries can be assigned. This set contains instances for several types of constraints.

At this presearch stage of the algorithm, the constraint (see Section IV) that has the largest impact is the "Fulfill Workload Constraint"—lorry assigned to job must be adequately equipped for the job. There is one "Fulfill Workload Constraint" per variable. The effect of posting each "Fulfill Workload Constraint" is that the domain of that variable will be guaranteed to contain only orders it can fulfil. This is done through domain reduction and is performed by the *Propagate* procedure prior to calling *Search*. All orders that the lorry is not equipped to fulfil are removed from its domain. If the problem is "over constrained," some of the resulting domains will be empty. When this happens, there is no solution and the algorithm exits. Otherwise, after the initial search space has been reduced, CLOE simply initialises the set of committed labels, i.e., the solution set, to be null and starts the actual search.

The *Search* procedure (Fig. 2) is the recursive algorithm that performs the actual search and is defined as follows. We will use the following additional notations:

x	an unassigned variable (an unassigned crane-lorry);
D_x	domain of variable x (set of all orders that that lorry can fulfil);

```
PROCEDURE SelectMostVersatileLorryFirst(U){
BEGIN
  versatility←0;
  lorry←NIL;
  REPEAT
    x←FIRST(U);
    v←ComputeVersatility(x);
    IF (v>versatility)
    THEN BEGIN
      lorry←x;
      versatility←v;
    END
  UNTIL (U={});
  RETURN lorry;
END
```

Fig. 3. SelectMostVersatileLorryFirst procedure.

$\langle x, v\rangle$	assignment of value v to variable x (the assignment of an order to a lorry);
U	set of unlabelled variables (unassigned crane-lorries);
CL	set of committed labels (assigned crane-lorries).

The main tasks for *Search* is to select the next best unassigned lorry using the *SelectLorry* procedure (Fig. 3), then assign the next best order with *SelectOrder* and finally, to *Propagate* constraints. *SelectLorry* defines the heuristic for selecting the next "best unassigned crane-lorry." *SelectOrder* defines the heuristic for selecting the next "best order to assign." We have experimented with different definitions and combinations of *SelectLorry* and *SelectOrder* to achieve better than human performance. These heuristics will be described later. *Propagate* is of course the CP algorithm that performs domain reduction. It is a type of look-ahead strategy [28] to detect unsatisfiability.

The key constraints (see Section IV) that are propagated during the search are instances of "No Overlap Constraint"—each lorry should only be assigned to one order at a time. There is one "No Overlap Constraint" per variable, which represents a lorry-to-order assignment. The effect of posting each "No Overlap Constraint" is that whenever an order has been assigned a lorry, all the domains of the remaining unassigned lorries will have their domains reduced by removing this order, as the order will no longer be a valid choice.

Whenever any domain gets reduced to an empty set or when a legal lorry-order assignment cannot be found, the *Search* algorithm returns NIL. This forces the algorithm to backtrack to a previous level and continues the search with other choice points.

B. SelectLorry Procedure

The *SelectLorry* procedure defines the heuristic for what next "best unassigned crane-lorry" means. We have experimented with many different heuristic definitions to find the one that gets the best improvement in the "quality" of the generated plans. Three of them are described in this paper. These heuristics are based on either "versatility," "size," or "experience" of crane-lorry-driver combination.

1) Versatility Heuristic: By selecting the most "versatile" lorry first, and then assigning the best order to it, we try to maximize customer satisfaction by assigning the "most preferred" equipment first and also maximize equipment utilization, which satisfies one of our design objectives (see Section IV): "maximize utilization"—minimize the number of orders given to subcontractors.

```
PROCEDURE SelectLargestLorryFirst(U){
BEGIN
  size←0;
  lorry←NIL;
  REPEAT
    x←FIRST(U);
    s←ComputeSize (x);
    IF (s>size)
    THEN BEGIN
       lorry←x;
       size←s;
    END
  UNTIL (U={});
  RETURN lorry;
END
```

Fig. 4. SelectLargestLorryFirst procedure.

The following is a simple procedure that defines this heuristic. "U" is the set of unassigned lorries. This procedure returns a lorry from this set that is most "versatile."

The *ComputeVersatility* procedure returns a number that defines the degree of "versatility" a crane-lorry has. The exact meaning of this measurement may vary from company to company. We designed CLOE such a way that this definition can easily be changed be adjusting parameters or weights in a database table. For the company we did this work on, "versatility" is defined as

$$V_x=(w_1 \times \mathrm{TM}_x)+(w_2 \times \mathrm{VW}_x)+(w_3 \times \mathrm{VA}_x)+(w_4 \times M_x)$$

where

V_x versatility of crane-lorry x;
TM_x normalized maximum ton·meters of the crane;
VW_x normalized maximum GVW of the lorry;
VA_x normalized payload area of the vehicle,
M_x mounting location of crane—rear is more versatile;
w_i weighting factor (varied for different companies).

The values are normalized to make setting of weights and comparisons easier to perform.

2) Size Heuristic: The "size" heuristic is a simplified form of the "versatility" heuristic and focuses just on the workload handling capability of the lorry. "Size" is defined to be the normalized maximum GVW of the lorry

$$S_x = \mathrm{VW}_x$$

where

S_x "size" of crane-lorry x;
VW_x normalized maximum GVW of the lorry.

By selecting the lorries that can carry more goods first, we try to maximize the number of larger orders that can be satisfied. In other words, when subcontracting is needed we can minimize subcontracted order size and satisfy another design objective (as described in Section IV): "Minimize Subcontract Order Size"—when subcontracting, subcontract orders that require smaller sized lorry to reduce cost.

The pseudocode for *SelectLargestLorryFirst* (Fig. 4) is also quite straightforward. Again, "U" is the set of unassigned lorries and the procedure returns a lorry from this set that is "largest" in size.

3) Experience Heuristic: The third heuristic is based on the "experience" of the crane-lorry-driver combination in performing similar tasks in the past and the feedback received from the customer on how "satisfied" they were with the service provided. This is computed using a set of historical data on previous crane-lorry assignments. Again the definition of "experience" may differ from company to company. For this company, "experience" is defined as

```
PROCEDURE SelectMostExperiencedLorryFirst (U){
BEGIN
  correlation←ComputeHighestCorrelation(U);
  Lorries←FindLorriesWithSameCorrelation(U, correlation);
  lorry←NIL;
  IF (only one element in Lorries) THEN lorry←FIRST(Lorries);
  ELSE lorry←SelectLargest LorryFirst (Lorries);
  RETURN lorry;
END
```

Fig. 5. SelectMostExperiencedLorryFirst procedure.

$$E_{\mathrm{cxj}} = (w_1 \times F_{\mathrm{cxj}} \times \mathrm{CLD}_{\mathrm{cxj}}) + (w_2 \times F_{\mathrm{cj}} \times \mathrm{CL}_{\mathrm{cj}}) + (w_3 \times F_{\mathrm{xj}} \times D_{\mathrm{xj}})$$

where

E_{cxj} "experience"—correlation of crane-lorry c and driver x to job type j ("job type" is defined in terms of location and job nature);
$\mathrm{CLD}_{\mathrm{cxj}}$ normalized number of previous assignments of crane-lorry-driver to j;
$\mathrm{CL}_{\mathrm{cj}}$ normalized number of previous assignments of crane-lorry only to j (excluding cases from CLD),
D_{xj} normalized number of previous assignments of driver only to j (excluding cases from CLD),
F_{cxj} average customer feedback on performance of crane-lorry-driver for job j;
F_{cj} average customer feedback on performance of crane-lorry for job j;
F_{xj} average customer feedback on performance of driver for job j;
w_i a weighting factor (varied for different companies).

The pseudocode for *SelectMostExperiencedLorryFirst* (Fig. 5) is as follows. It uses the "size" heuristic, i.e., the *SelectLargestLorryFirst* procedure, as its secondary selection criteria.

The "experience" heuristic was designed to satisfy the following design objective (see Section IV): "maximize experience"—most experienced lorry-driver should be assigned, whenever possible.

4) The SelectOrder Procedure: The *SelectOrder* procedure defines the heuristic for what next "best order to assigned" means. We have used the following heuristic in our three test algorithms to define "best order": "Always select the largest orders a lorry can handle and from these the order that it has most experience in handling."

The pseudocode that encodes this heuristic is as follows. "Lorry" is the unassigned lorry that was selected previously by one of the *SelectLorry* heuristics. "$\mathrm{D}_{\mathrm{lorry}}$" is the domain of this lorry, i.e., a set of orders it can fulfill. The procedure selects one order from this set that requires the "largest" lorry, i.e., order that has most stringent requirements on lorry capabilities, and in which this lorry has most "experience" in handling (Fig. 6).

The definition of "largest" may vary from company to company. This may be in terms of maximum GVW, in terms of the maximum capacity the crane can handle, the size of the lorry, or combinations of these factors. In the company we performed

```
PROCEDURE MaximizeUtilization (lorry, D_lorry){
BEGIN
  size←FindLargestLorryNeeded(D_lorry);
  Orders←FindOrdersWithSameLorrySize(D_lorry, size);
  order←NIL;
  IF (only one element in Orders) THEN order←FIRST(Orders);
  ELSE order←MaximizeCorrelation(lorry, Orders);
  RETURN order;
END
```

Fig. 6. MaximizeUtilization procedure.

this work on, we define "large" as the following:

$$L_x = (w_1 * \mathrm{TM}_x) + (w_2 * \mathrm{VW}_x) + (w_3 * \mathrm{VA}_x)$$

where

L_x "largeness" of crane-lorry x;
TM_i normalized maximum ton $\cdot$ meters of the crane,
VW_x normalized maximum GVW of the lorry,
VA_x the normalized payload area of the vehicle,
w_i weighting factor (varied for different companies).

VIII. Test Results

In this paper, we will show results from three of the CLOE algorithms we have tested. Each algorithm executes the same *Search* procedure as defined in Section VII, but with a different set of heuristics:

- CLOE Algorithm 1 (versatility).
  - *Select Lorry Heuristic:* SelectMostVersatileLorry-First.
  - *Select Order Heuristic:* MaximizeUtilization.
- CLOE Algorithm 2 (size).
  - *Select Lorry Heuristic:* SelectLargestLorryFirst.
  - *Select Order Heuristic:* MaximizeUtilization.
- CLOE Algorithm 3 (experience).
  - *Select Lorry Heuristic:* SelectMostExperienced LorryFirst.
  - *Select Order Heuristic:* MaximizeUtilization.

These three algorithms were tested on seven sets of actual-order data. Historical data, used to determine "experience," was compiled using the prior two weeks worth of actual plans as well as customer feedback on driver performance. Results from testing these three algorithms were then compared according to two key criteria—"cost" and "quality." "Cost" is measured by the number and types of vehicles that must be outsourced to subcontractors, as this represents actual additional cost that the company must pay out. "Quality" is measured mainly in terms of customer satisfaction, and this translates to the number of experienced vehicles that were assigned, in which customers were happy with in the past. The company-specific weighting factors that are used in formulas to compute the "versatility," "experience," and "largeness" were determined through interviews with the human schedulers, and then fixed for all the experiments described in this section.

A. Cost Factor

In evaluating our algorithms compared with manual approach, we are most interested in the additional cost that the company must bear. This is measured in terms of the number and type of vehicles that are outsourced to other subcontractors. Vehicle type is important, as "larger" vehicles will be more expensive to outsource. For simplicity, we represent the vehicle type as the maximum lifting capability of the mounted crane, in terms of tons of goods it can lift. We assume the crane outreach is 2 m. For example, in our tables, "25T" means a vehicle with a crane that can lift 25 ton of goods with an outreach of 2 m, i.e., the vehicle has a 50-ton·m crane.

TABLE I
Outsourced Vehicle Tonnage and Cost Savings

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Manual	175	170	127	161	140	123	85
CLOE	170	162	127	161	137	125	85
%Reduction	2.86%	4.71%	0.00%	0.00%	2.14%	-1.63%	0.00%

From our test cases, the three variants of the CLOE algorithm, all performed equally well in terms of outsourced vehicles. Table I shows the "total tonnage" that was outsourced for the manual and CLOE plans. The "total tonnage" is a rough estimation of the cost of outsourcing, as larger tonnage vehicles will be more expensive to lease. The table also shows the cost savings in the form of "percentage reduction" in outsourcing cost due to CLOE optimization.

In all our test cases, CLOE performed better than the manual approach and reduced overall cost, except for Test 6. After comparing results of the manual and CLOE generated plans for Test 6, we found that the manual plan was in fact incorrect and that a smaller crane-lorry was mistakenly assigned to one of the orders, which did not quite meet customer specifications! This is precisely the kind of errors that can easily be prevented in an automated approach, such as our system.

B. Customer Satisfaction Factor

The second key criterion for our assignment algorithm is whether it can generate assignments that customers will be happy with. "Customer Satisfaction" for our crane-lorry assignment problem is measured in terms of the number of assignments that allocated "experienced" crane-lorry-driver combination that the customer was satisfied with in the past. To simplify comparisons, we classified each assignment into one of six categories—very high, high, moderate, average, low, or very low customer satisfaction. Assignments are classified by comparing them to historical assignment statistics for a particular customer. For example, a new assignment would fall into the "very high" or "high" category if the same crane-lorry-driver combination has previously been assigned to service the same type of job for the same customer on most occasions and the customer feedback was positive. Conversely, a new assignment would get a "very low" rating if the crane-lorry-driver combination has never been used before to service this customer. Fig. 7 summarizes the number of assignments made in each classification. As the figure shows, Algorithm 3 (experience) allocates more assignments with high customer satisfaction and fewer assignments with low customer satisfaction that other algorithms or the manual approach. Algorithm 3 (experience), on average, assigns 18% more assignments with high customer satisfaction

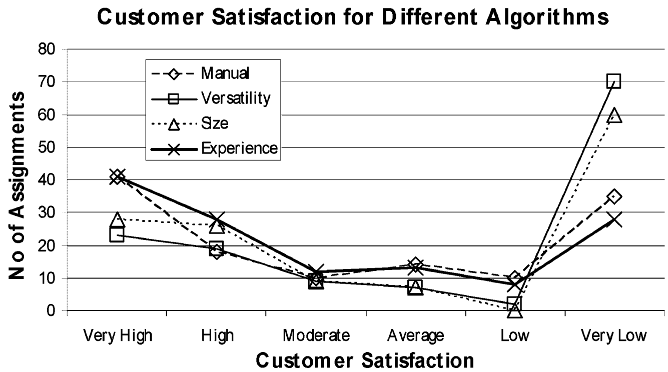

Fig. 7. Degree of customer satisfaction.

TABLE II
PERCENTAGE OF "VERY HIGH" PLUS "HIGH" CUSTOMER SATISFIED ASSIGNMENTS AND THE PERCENTAGE IMPROVEMENTS FOR EACH ALGORITHM COMPARED WITH MANUAL APPROACH

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Manual	53%	44%	42%	44%	41%	47%	50%
A1: Versatility	32%	32%	37%	39%	29%	32%	26%
A2: Size	37%	37%	42%	44%	47%	42%	42%
A3: Experience	53%	53%	53%	56%	53%	53%	53%
%Improve (A1)	-40%	-29%	-13%	-13%	-29%	-33%	-47%
%Improve (A2)	-30%	-17%	0%	0%	**14%**	-11%	-16%
%Improve (A3)	0%	**18%**	**25%**	**25%**	**29%**	**11%**	**5%**

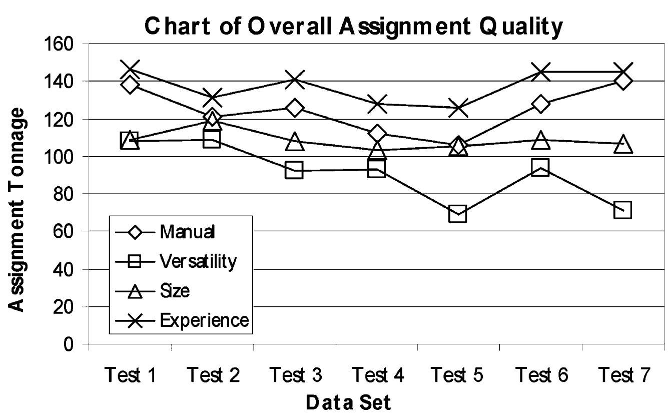

Fig. 8. Overall quality of generated plan.

compared with manual approach. This is a significant improvement in achieving customer satisfaction.

Table II shows the percentage of "very high" plus "high" customer satisfied assignments made by the three algorithms compared with the manual approach. On average, Algorithm 3 (experience) provides 16% improvement in increasing the percentage of "very high" plus "high" customer satisfied assignments in the plan generated.

C. Overall Quality

Fig. 8 shows the overall quality of the plan generated for each of the seven test cases. The "quality" measure considers both the "cost" and "customer satisfaction" factors, i.e., the ability to provide quality service while maximizing revenue. We define this to be the total tonnage of all assignments that fall within the "very high" and "high" customer satisfaction category. A larger number means more customers are satisfied and less outsourcing is needed. The figure shows that Algorithm 3 (experience) performs significantly better than the manual approach. Although Algorithm 1 (versatility) and Algorithm 2 (size) performs just as well as Algorithm 3 (experience) in our previous "cost" comparison, they do not perform well in the "quality" comparison. This is due purely to the definition of "quality" that is used by the company. For our test case, even though there is a dual objective of minimizing "cost" while maximizing "customer satisfaction," the second objective of "customer satisfaction" is more important to the company involved. The "experience" element is one of the key factors in maximizing customer satisfaction and, hence, Algorithm 3 (experience) is at an advantage. For other crane-lorry companies that may have different quality measurements, Algorithm 1 or 2 might become more interesting.

IX. CONCLUSION

The quality of service provided by a company is very important to service-oriented companies, such as lorry leasing. This is the main differentiating factor between itself and other competitors in the region. Being able to produce a high-quality plan means better services are provided and is the key design objective of our assignment system. Our testing shows that plans generated by CLOE are better than those produced by human planners and, hence, improve quality of service. At the same time, plans are produced within seconds compared to close to an hour if done manually. All necessary constraints and criteria are considered systematically and results are automatically forwarded through our B2B exchange to other business systems for processing, such as billing. We have experimented with many different types of search heuristics and have analyzed their effects on plan quality. We have found that by considering the experience of the crane-lorry-driver combination and by maximizing on the order assignment, we were able to generate plans that were significantly better than those produced by human planners, and within a substantially shorter time. Although the use of constraint-based assignment techniques is still limited in the vehicle leasing industry, we hope our combination of automated assignment with Internet portal technology that streamlines B2B, B2C, and B2E communications can excite interest in this area.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manag. Sci.*, vol. 6, pp. 80–91, 1959.
[2] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, pp. 568–581, 1964.
[3] J.-F. Puget, "Object-oriented constraint programming for transportation problems," in *Proc. Inst. Electr. Eng. Colloq. Adv. Softw. Technol. Scheduling*, Apr. 1993, pp. 411–413.
[4] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: Univ. Michigan Press, 1976.
[5] J.-Y. Potvin, Y. Shen, and J.-M. Rousseau, "Neural networks for automated vehicle dispatching," *Comput. Oper. Res.*, vol. 19, no. 3/4, pp. 267–276, 1992.
[6] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, vol. 1520, 1998, pp. 417–431.

[7] Y. Caseau and F. Laburthe, "Disjunctive scheduling with task intervals," Lab. d'Inf. de l'Ecole Normale Supérieure, Paris, France, LIENS Tech. Rep. 95-25, Jul. 1995.
[8] Y. Caseau and F. Laburthe, "Solving small TSPs with constraints," in *Proc. 14th Int. Conf. Logic Program.*, 1997, pp. 316–330.
[9] L. S. Ochi, D. S. Vianna, L. M. A. Drummond, and A. O. Victor, "A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet," *Future Gener. Comput. Syst.*, vol. 14, pp. 285–292, 1998.
[10] N. Barnier and P. Brisset, "Optimization by hybridization of a genetic algorithm with constraint satisfaction techniques," in *Proc. IEEE World Congr. Comput. Intell.—Evol. Comput.*, 1998, pp. 645–649.
[11] P. W. Frizzell and J. W. Giffin, "The split delivery vehicle scheduling problem with time windows and grid network distances," *Comput. Oper. Res.*, vol. 22, no. 6, pp. 655–667, 1995.
[12] J. S. Shang and C. K. Cuff, "Multicriteria pickup and delivery problem with transfer opportunity," *Comput. Ind. Eng.*, vol. 30, no. 4, pp. 631–645, 1996.
[13] M. Desrochers, C. V. Jones, J. K. Lenstra, M. W. P. Savelsbergh, and L. Stougie, "Towards a model and algorithm management system for vehicle routing and scheduling problems," *Decision Supp. Syst.*, vol. 25, pp. 109–133, 1999.
[14] F. Baita, R. Pesenti, W. Ukovich, and D. Favaretto, "A comparison of different solution approaches to the vehicle scheduling problem in a practical case," *Comput. Oper. Res.*, vol. 27, pp. 1249–1269, 2000.
[15] Y. B. Park and S. H. Song, "Vehicle scheduling problems with time-varying speed," *Comput. Ind. Eng.*, vol. 33, no. 3–4, pp. 853–856, 1997.
[16] Y. B. Park, "A solution of the bicriteria vehicle scheduling problem with time and area-dependent travel speeds," *Comput. Ind. Eng.*, vol. 38, pp. 173–187, 2000.
[17] G. M. Campbell and M. Diaby, "Development and evaluation of an assignment heuristic for allocating cross-trained workers," *Eur. J. Oper. Res.*, vol. 138, pp. 9–20, 2002.
[18] T. Sawik, "A multilevel machine and vehicle scheduling in a flexible manufacturing system," *Math. Comput. Model.*, vol. 23, no. 7, pp. 45–57, 1996.
[19] G. Ulusoy, F. S. Serifoglu, and U. Bilge, "A genetic algorithm approach to the simultaneous scheduling of machines and automated guided vehicles," *Comput. Oper. Res.*, vol. 24, no. 4, pp. 335–352, 1997.
[20] X. B. Wu, "The application of analytic process of resource in an AGV scheduling," *Comput. Ind. Eng.*, vol. 35, no. 1–2, pp. 169–172, 1998.
[21] X. Huang and J. Hallam, "Spring-based negotiation for conflict resolution in AGV scheduling," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.,* Oct. 22–25, 1995, vol. 1, pp. 789–794.
[22] J. B. H. Kwa, "Tolerant planning and negotiation in multi-agent environments," *Appl. Artif. Intell.*, vol. 2, no. 3–4, pp. 179–211, 1988.
[23] ——, "Planning automated guided vehicle movements in a factory," Ph.D. dissertation, Univ. Edinburgh, Edinburgh, 1988.
[24] A. J. Gill, "SUNRAY V—An intelligent container trucking management system," presented at the 1st ILOG SOLVER and ILOG SCHEDULE Int. User's Meeting, Abbaye des Vaux de Cernay, France, Jul. 1995.
[25] T. Duncan, "Schedule-IT: An intelligent vehicle scheduling," presented at the 1st ILOG SOLVER and ILOG SCHEDULE Int. User's Meeting, Abbaye des Vaux de Cernay, France, Jul. 1995.
[26] H. W. Chun, "Constraint programming in Java with Jsolver," presented at the 1st Int. Conf. Exhib. PACLP, London, Apr. 1999.
[27] Applied Artificial Intelligence. (2001, May). CSIRO Mathematical and Information Sciences. [Online]. Available: http://www.cmis.csiro.au/aai/
[28] E. Tsang, *Foundations of Constraint Satisfaction*. San Diego, CA: Academic, 1993.
[29] J. Cohen, "Constraint logic programming," *Commun. ACM*, vol. 33, no. 7, pp. 52–68, Jul. 1990.
[30] G. L. Steele, Jr., "The definition and implementation of a computer programming language based on constraints," Ph.D. dissertation, MIT, Cambridge, MA, 1980.
[31] V. Kumar, "Algorithms for constraint satisfaction problems: A survey," *AI Mag.*, vol. 13, no. 1, pp. 32–44, 1992.
[32] P. Van Hentenryck, *Constraint Satisfaction in Logic Programming*. Cambridge, MA: MIT Press, 1989.
[33] H. W. Chun, "Scheduling as a multi-dimensional placement problem," *Eng. Appl. Artif. Intell.*, vol. 9, no. 3, pp. 261–274, Jun. 1996.
[34] H. W. Chun and R. W. T. Mak, "Intelligent resource simulation for an airport check-in counter allocation system," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 29, no. 3, pp. 325–335, Aug. 1999.
[35] H. W. Chun, S. Chan, F. Tsang, and D. Yeung, "Stand allocation system (SAS): A constraint-based system developed with software components," *AI Mag.*, vol. 21, no. 4, pp. 63–74, 2000.
[36] D. Frost and R. Dechter, "Look-ahead value ordering forconstraint satisfaction problems," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, 1995, vol. 1, pp. 572–578.
[37] A. K. Mackworth, "Consistency in networks of relations," *Artif. Intell.*, vol. 8, no. 1, pp. 99–118, 1977.
[38] A. K. Mackworth and E. C. Freuder, "The complexity of some polynomial network consistency algorithms for constraint satisfaction problems," *Artif. Intell.*, vol. 25, pp. 65–74, 1985.

Andy Hon Wai Chun (S'79–M'81) received the B.S. degree from the Illinois Institute of Technology, Chicago, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, in 1983 and 1987, respectively.

He was a Senior Scientist at Ascent Technology, Boston, MA, before joining the Information Engineering Department, Chinese University of Hong Kong as Lecturer in 1989. In 1994, he joined the City University of Hong Kong, Hong Kong, China, where he is currently an Associate Professor in the Department of Computer Science. He is a Consultant for the CityU Professional Services Limited, Hong Kong. His current research interests include AI scheduling, optimization, and resource allocation.

Rebecca Y. M. Wong received the B.Sc. degree in information technology and the M.Phil. degree in electronic engineering from the City University of Hong Kong, Hong Kong, China, in 1999 and 2001, respectively.

After obtaining the M.Phil. degree, she joined an industrial company to help develop software systems for manufacturing and office automation. Her research interests are artificial intelligence, optimization, and office automation. She is currently a Software Developer with the Sun Huang Kai Financial Group, Queensway, Hong Kong.