CWI

Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

M. Desrochers, J.K. Lenstra, M.W.P. Savelsbergh

A classification scheme for vehicle routing and scheduling problems

Department of Operations Research and System Theory Report OS-R8721 November

A Classification Scheme for Vehicle Routing and Scheduling Problems

M. Desrochers
Centre for Mathematics and Computer Science, Amsterdam
Ecole des Hautes Etudes Commerciales, Montreal

J.K. Lenstra
M.W.P. Savelsbergh
Centre for Mathematics and Computer Science, Amsterdam
Erasmus University, Rotterdam

We propose a classification scheme for a class of models that arise in the area of vehicle routing and scheduling and illustrate it on a number of problems that have been considered in the literature. The classification scheme may serve as a first step towards the development of a model and algorithm management system in this area.

1980 Mathematics Subject Classification: 90B05, 90B35, 68T30.
Key Words & Phrases: classification, routing, scheduling, model, algorithm.
Note: This paper has been submitted for publication.

1. INTRODUCTION

During the last twenty years, many papers have been devoted to the development of optimization and approximation algorithms for vehicle routing and scheduling problems. This interest is due to the practical importance of effective and efficient methods for handling physical distribution situations as well as to the intriguing nature of the underlying combinatorial optimization models. The great variety of vehicle routing and scheduling problems in practice and the large number of existing algorithms make it difficult for an unexperienced distribution manager, and even for an experienced one, to select a method that is well suited for his specific situation. In order to facilitate this decision process, we propose to develop a system that provides support in modeling problem situations and in suggesting algorithms that might be applicable to the resulting models.

The system will represent and manipulate information at three different levels. At the first level, there is the *real-life problem situation*. It may contain many aspects that are not relevant for the selection of a solution method. At the second level, there is the *abstract problem type*. It is obtained from the real-life problem situation by determining and modeling the relevant entities that describe it in terms of decisions, objectives and constraints. At the third level, there are the *algorithms*. One that appears to be suitable in the situation at hand is selected.

The knowledge and expertise that must be built into the system concern two different issues. On the one hand, there is the knowledge and expertise that is applied to obtain an abstract representation of the problem situation. On the other hand, there is the knowledge and expertise that is applied to choose from among the multitude of available algorithms one that is appropriate for this model.

There is a vast literature on vehicle routing and scheduling that contains the knowledge and expertise of either type. This is one interesting aspect of the project: the knowledge exists, and the question is how to formalize its use. In order to meet this challenge, we will have to create a vocabulary for representing the knowledge and to design inference algorithms for manipulating it.

It is often difficult to keep track of all the available information on a problem class, even if the class is

Report OS-R8721
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

well structured and the information is of very elementary kind. Lageweg, Lenstra, Lawler, and Rinnooy Kan [1981, 1982] built a specialized inference engine in order to be able to keep track of the complexity results for a class of 4536 deterministic machine scheduling problems. Their main purpose was to determine the complexity status of each of these problems: solvable in polynomial time, *NP*-hard, or open. The resulting MSPCLASS system, using simple inference rules and straightforward knowledge on problem transformations, is able to deduce listings of essential results: maximal easy problems, minimal and maximal open problems, and minimal hard problems. To construct these listings by hand would be a very tedious task.

While the system we propose to develop calls for the synthesis of a massive amount of knowledge on vehicle routing and scheduling, achieving such a synthesis is already a worthwhile purpose in itself. We hope that the classification scheme which is presented in this paper is a step in this direction.

There already exist classification schemes for other problem areas in operations research. Conway, Maxwell, and Miller [1967] introduced a four-parameter notation to classify deterministic and stochastic scheduling problems. Graham, Lawler, Lenstra, and Rinnooy Kan [1979] extended and modified this system for the class of deterministic machine scheduling problems; their scheme formed the basis for the specialized inference engine mentioned above. Handler and Mirchandani [1979] classified a limited class of location problems. Bodin and Golden [1981] outlined a classification scheme for vehicle routing and scheduling problems, and Ronen [1987] lists a number of practical problem characteristics in this area.

In Section 2, we propose a language to define abstract problem types in vehicle routing and scheduling. We have tried to limit the vocabulary of this language, but the size of the problem class under consideration forced us to compromise on several points. In Section 3, the use of the language is illustrated on a number of examples.

2. THE DEFINITION LANGUAGE

A number of vehicles, stationed at one or more depots, have to serve a collection of customers in such a way that given constraints are respected and a given objective function is optimized. To define one such problem type in a formal way, our language uses four fields. The first field describes the characteristics and constraints that are relevant only to single addresses (customers and depots). We prefer the term 'address' to 'customer' because of the great variety of customer types: apart from the usual single-address customer, there is also the customer corresponding to an origin-destination pair or to all the addresses located on a street segment. The second field specifies the characteristics relevant only to single vehicles. The third field contains all problem characteristics that cannot be identified with single addresses or vehicles. The fourth field defines one or more objective functions.

A fifth field may be added to describe additional information about a specific class of problem instances. Although such information does not belong to the model as defined in our four fields, it might still be useful for the selection of a suitable algorithm. For example, it might be helpful to know the average number of addresses that are to be assigned to one vehicle. The specification of this field has been postponed until the development of the system is at a more advanced stage.

All the elements in our problem definition are in principle unidimensional. However, superscripts can be added to indicate multidimensional constraints. For example, cap_i indicates that vehicle load is bounded from above in one dimension, which may be, e.g., volume or weight; cap_i^2 indicates a two-dimensional capacity constraint, which may mean that upper bounds on both volume and weight are to be taken into account.

The classification language consists of a set of rules that define allowable structures. Each rule defines a nonterminal symbol in terms of other nonterminal symbols (fields, subfields, and elements) and terminal symbols (values of elements, or 'tokens'); the symbol $\vee$ is used to represent an exclusive or. Each nonterminal symbol is enclosed in angular brackets. Each token is followed by a comment on its interpretation between square brackets. The token $\circ$ indicates the empty symbol; it is used to indicate a default value, which is usually either the simplest or the most frequently occurring value.

Each problem type in the class under consideration is defined by a number of tokens, some of which may be equal to $\circ$. For notational convenience, two successive tokens are separated by a vertical bar if they belong to different fields and by a comma if they belong to the same field and are both not equal to ∘.

Note that we have chosen for a brief verbal interpretation of each token rather than for a complete definition in mathematical terms. A formal approach, while possible and useful in itself, would distract our attention from the main purpose of this paper.

<classification> ::=

<addresses>
<vehicles>
<problem characteristics>
<objectives>

2.1. *Addresses*

The first field defines the characteristics that can be associated with single addresses. All the addresses will be located on a network $G = (V,E)$ with a set V of nodes and a set E of (undirected) edges and (directed) arcs. There are four subfields.

The first subfield specifies the *number of depots*. There are single-depot problems and problems where the number of depots is given as part of the problem instance.

The second subfield specifies the *type of demand*. There are three parts. First the location of the demand: ∘ indicates that the customers are located on the nodes, EDGE indicates that the customers are located on the edges (arcs) of the network, MIXED indicates that the customers are located on both the nodes and edges (arcs), and TASK indicates the case that each customer corresponds to an origin-destination pair; the load is picked up at the origin address and delivered to the destination address. The second part of the subfield specifies if all the demands are of the same type (all deliveries or all collections) or not (mixed deliveries and collections). The third part specifies the nature of the demand: deterministic or stochastic.

The third subfield specifies the *address scheduling constraints*, i.e., the temporal aspect of the demand. Either there are no temporal constraints, or the departure time is fixed (fixed schedule), or the departure time is restricted to a single interval (single time windows) or to a set of intervals (multiple time windows).

The final subfield specifies the *address selection constraints*. There is a basic distinction between two problem classes. In the first class, a single plan is to be made for the given collections of addresses and vehicles. There are three subclasses: all addresses must be visited; a given subset of addresses must be visited and the others may be visited if it is profitable; or the addresses are partitioned into subsets and at least one address in each subset must be visited. In the second class, a number of plans is to be made over a certain time period, during which the addresses must be visited with given priorities or at given frequencies. This requirement is open to various specifications. It may lead to problems with a longer time horizon, in which a weekly allocation problem is to be solved before the daily routing problem is defined.

<addresses> ::=

<number of depots>
<type of demand>
<address scheduling constraints>
<priority constraints>

<number of depots> ::= $1 \vee l$

1	[one depot]
l	[specified as part of the problem instance]

<type of demand> ::= <α_1> <α_2> <α_3>

$\langle\alpha_1\rangle ::= \circ \vee \text{EDGE} \vee \text{MIXED} \vee \text{TASK}$

$\circ$	[node routing]
EDGE	[edge routing]
MIXED	[mixed routing (nodes and edges)]
TASK	[task routing]

$\langle\alpha_2\rangle ::= \circ \vee \pm$

$\circ$	[either all deliveries or all collections]
$\pm$	[mixed deliveries and collections]

$\langle\alpha_3\rangle ::= \circ \vee \sim$

$\circ$	[deterministic demand]
$\sim$	[stochastic demand]

<address scheduling constraints> $::= \circ \vee fs_j \vee tw_j \vee mw_j$

$\circ$	[no scheduling constraints]
fs_j	[fixed schedule]
tw_j	[single time windows]
mw_j	[multiple time windows]

<address selection constraints> $::= \circ \vee$ *subset* $\vee$ *choice* $\vee$ *period*

$\circ$	[single plan; all addresses must be visited]
subset	[single plan; a given subset of addresses must be visited]
choice	[single plan; at least one address in each subset of a given partition must be visited]
period	[a number of plans over a given time period is to be made]

2.2. *Vehicles*

The second field defines the characteristics of the vehicles and their routes. There are three types of information in this field: the number of vehicles, physical characteristics of the vehicles, and temporal constraints on a route.

The first subfield specifies the *number of vehicles*: the number of vehicles is a constant, specified as part of the problem type, or a variable, specified as part of the problem instance. The symbol '=' can be used to indicate that all vehicles must be used.

The second and third subfields specify the *physical characteristics* of the vehicles: the capacity and the presence of compartments. The fleet can be *homogeneous* (all vehicles have the same capacity) or *heterogeneous*. There are two types of compartmentalized vehicles. Some vehicles have *interchangeable* compartments. These can be used to separate incompatible commodities such as chickens and foxes. Other vehicles have *dedicated* compartments, each used to store one type of good; e.g., frozen meals and fresh vegetables must be kept in separate dedicated compartments.

The fourth and fifth subfields specify *temporal constraints*. There can be availability intervals for the vehicles and upper bounds on the duration of the routes.

<vehicles> ::=

<number of vehicles>
<capacity constraints>

<commodity constraints>
<vehicle scheduling constraints>
<route duration constraints>

<number of vehicles> ::= $<\beta_1> <\beta_2>$

$<\beta_1>$ ::= $\circ \vee =$

$\circ$	[at most β_2 vehicles can be used]
$=$	[all β_2 vehicles must be used]

$<\beta_2>$ ::= $c \vee m$

$c\ (c \in \mathbb{N})$	[c vehicles]
m	[specified as part of the problem instance]

<capacity constraints> ::= $\circ \vee cap \vee cap_i$

$\circ$	[no capacity constraints]
cap	[vehicles with identical capacities]
cap_i	[vehicles with different capacities]

<commodity constraints> ::= $\circ \vee sep \vee ded$

$\circ$	[no compartments]
sep	[vehicles have interchangeable compartments]
ded	[vehicles have dedicated compartments]

<vehicle scheduling constraints> ::= $\circ \vee tw \vee tw_i$

$\circ$	[no scheduling constraints]
tw	[identical time windows for vehicles]
tw_i	[different time windows for vehicles]

<route duration constraints> ::= $\circ \vee dur \vee dur_i$

$\circ$	[no route duration constraints]
dur	[identical upper bounds on route duration]
dur_i	[different upper bounds on route duration]

2.3. *Problem characteristics*

The third field defines the network underlying the problem, the service strategy, and constraints on the relations between addresses and vehicles.

The first subfield specifies the properties of the *network* (directed, undirected or mixed) and of the *travel time matrix* (satisfying the triangle inequality or not).

The second subfield specifies the *service strategy* chosen by the user. There are four types of strategic decisions.

(1) The first type allows or disallows splitting of the customer demand. A priori splitting of demand occurs when it is decided at the outset that the demand may be satisfied by more than one visit to the customer. A posteriori splitting of the demand occurs in the case of stochastic demand when, once on the

customer premises, the driver discovers that the demand is larger than foreseen and decides not to satisfy the demand completely during that visit.

(2) In the case of node routing with mixed deliveries and collections, the user can choose for *backhauling*, i.e., delivering first to empty the vehicle and then collecting loads on the way back to the depot. In the case of task routing, the user can choose for *full load* routing, i.e., only one load can be in the vehicle at any time.

(3) In most cases a vehicle performs at most one route per period, but the user can allow more than one route per vehicle.

(4) Usually vehicles are restricted to start and finish at the same depot, but this can be relaxed and the user can allow multi-depot routes.

The other subfields specifies the possible *relations* between two addresses, between an address and a vehicle, or between two vehicles. Such relations are caused by a number of very different factors, and enumerating these here would not be feasible. Instead of describing the underlying factors, we have chosen to specify the restrictions caused by them.

The best known of these relations is the *precedence constraint* between two customers: the vehicle must visit one customer before visiting the other. Note that these precedence constraints have nothing to do with the implicit precedence constraints in the origin-destination pairs in TASK routing, because we view such an origin-destination pair as a single customer.

Most of the other relations are *inclusion* and *exclusion restrictions*. It may be that an address must be served from a given depot, must be allocated to the same route as another address, or must be visited by a given vehicle. For example, an address-vehicle inclusion restriction occurs if the vehicle must be equipped with an unloading device because the customer has no delivery dock. It may also be that an address should *not* be served from a given depot, should *not* be allocated to the same route as another address, or should *not* be visited by a given vehicle.

The last type of restriction is *vehicle synchronization*, occurring when two or more vehicles must exchange loads or assist each other.

<problem characteristics> ::=

<type of network>
<type of strategy>
<address-address restrictions>
<address-vehicle restrictions>
<vehicle-vehicle restrictions>

<type of network> ::= $<\gamma_1> <\gamma_2>$

$<\gamma_1> ::= \circ \vee \Delta$

$\circ$	[general costs]
Δ	[the costs satisfy the triangle inequality]

$<\gamma_2> ::= \circ \vee dir \vee mix$

$\circ$	[undirected network]
dir	[directed network]
mix	[mixed network]

<type of strategy> ::= $<\delta_1> <\delta_2> <\delta_3> <\delta_4>$

$<\delta_1> ::= \circ \vee / \vee \div$

$\circ$	[splitting of demand not allowed]
$/$	[a priori splitting of demand allowed]
$\div$	[a posteriori splitting of demand allowed]

$<\delta_2> ::= \circ \vee \mathit{back} \vee \mathit{full}$

$\circ$	[no backhauling or full loads required]
back	[backhauling, in case of node routing]
full	[full loads, in case of task routing]

$<\delta_3> ::= \circ \vee \geqslant 1R/V$

$\circ$	[at most one route per vehicle]
$\geqslant 1R/V$	[more than one route per vehicle allowed]

$<\delta_4> ::= \circ \vee \geqslant 1D/R$

$\circ$	[a route starts and finishes at the same depot]
$\geqslant 1D/R$	[multi-depot routes allowed]

<address-address restrictions> $::= <\epsilon_1> <\epsilon_2> <\epsilon_3>$

$<\epsilon_1> ::= \circ \vee \mathit{prec}$

$\circ$	[no precedence constraints]
prec	[precedence constraints]

$<\epsilon_2> ::= \circ \vee DA$

$\circ$	[no depot-address restrictions]
DA	[depot-address restrictions]

$<\epsilon_3> ::= \circ \vee AA$

$\circ$	[no address-address restrictions]
AA	[address-address restrictions]

<address-vehicle restrictions> $:: = <\zeta_1> <\zeta_2>$

$<\zeta_1> ::= \circ \vee DV$

$\circ$	[no depot-vehicle restrictions]
DV	[depot-vehicle restrictions]

$<\zeta_2> ::= \circ \vee AV$

$\circ$	[no address-vehicle restrictions]
AV	[address-vehicle restrictions]

<vehicle-vehicle restrictions> ::= $\circ \vee VV$

$\circ$	[no vehicle-vehicle restrictions]
VV	[synchronization between vehicles needed]

2.4. *Objectives*

The fourth field defines the objective functions. To specify an objective function we introduce five quantities.

The travel and service time of vehicle i, i.e., the tour duration of its route, will be denoted by T_i. With this quantity we can express the standard objectives found in the vehicle routing and scheduling literature: minimization of the total travel and service time, and minimization of the span of a solution.

To be able to express more realistic objective functions, we introduce a vehicle cost function C_i, an address cost function c_j, a vehicle penalty function P_i, and an address penalty function p_j. A vehicle cost function can be used to model situations where, in addition to routing and scheduling, it is also required to determine the fleet size and mix. An address cost function allows the modeling of costs incurred due to deviation of a preferred service level. The penalty functions enable the modeling of costs incurred due to the violation of constraints. Sometimes it is allowed to violate constraints at a certain cost, if it is profitable; driver overtime is an example.

In practice, the problems often have a composite objective function. The user can specify this by listing the components of the objective function in order of decreasing importance. At the other end, our schemes also leaves the possibility open that no objective is specified, so that the problem reduces to a feasibility question.

As an addendum to the rules below, we note that, in the case of a single vehicle, the operator sum or max and the subscript i are dropped in the objectives related to routes and vehicles.

<objectives> ::= <objective> $\vee$ <objectives> <objective>

<objective> ::= $\circ \vee$ <operator> <function>

<operator> ::= sum $\vee$ max

sum	[minimize the sum of the cost function values]
max	[minimize the maximum cost function value]

<function> ::= $T_i \vee C_i \vee P_i \vee c_j \vee p_j$

T_i	[route duration]
C_i	[vehicle costs]
P_i	[vehicle penalty]
c_j	[address costs]
p_j	[address penalty]

3. Examples

In this section, fourteen problems taken from the vehicle routing and scheduling literature are classified using the scheme given in the previous section. This presentation has a twofold purpose. First, the examples illustrate the use of the classification scheme. Secondly, they show that a broad class of problems, including very practical ones, can be handled.

The examples are all presented in the same format. The difference in style of the various problem descriptions is due to the fact that we have quoted the source texts throughout.

Example 1. $1 | 1 | | T$ [Jaques 1859]

'In this new Game (invented by Sir William Rowan Hamilton, LL.D., &c., of Dublin, and by him named

Icosian, from a Greek word signifying "twenty") a player is to place the whole or part of a set of twenty numbered pieces or men upon the points or in the holes of a board ... in such a manner as always to proceed *along the lines* of the figure'. [The board is a planar representation of the pentagonal dodecahedron].

Example 2. $1\,|\,1\,|\,\Delta\,|\,T$ [Menger 1930]
'Wir bezeichnen als *Botenproblem* (weil diese Frage in der Praxis von jedem Postboten, übrigens auch von vielen Reisenden zur lösen ist) die Aufgabe, für endlichviele Punkte, deren paarweise Abstände bekannt sind, den kürzesten die Punkte verbindenden Weg zu finden.' [We call this the *messenger problem* (because in practice the problem has to be solved by every postman, and also by many travelers): finding the shortest path joining all of a finite set of points, whose pairwise distances are known.]

Example 3. 1,EDGE$\,|\,1\,|\,|\,T$ [Guan 1962]
'When the author was plotting a diagram for a mailman's route, he discovered the following problem: "A mailman has to cover his assigned segment before returning to the post office. The problem is to find the shortest walking distance for the mailman." This problem can be reduced to the following: "Given a connected graph in the plane, we are to draw a continuous graph (repetition permitted) from a given point and back minimizing the number of repeated arcs."'

Example 4. $1\,|\,m,cap\,|\,|\,\text{sum}T_i$ [Dantzig and Ramser 1959]
'The Traveling Salesman Problem may ... be generalized by imposing the condition that specified deliveries q_i be made at every point P_i (excepting the terminal point). If the capacity of the carrier $C \geqslant \Sigma_i q_i$, the problem is formally identical with the Traveling-Salesman Problem in its original form ... the Truck Dispatching Problem is characterized by the relation $C << \Sigma_i q_i$. ... For simplicity of presentation it will be assumed that only one product is to be delivered and that all trucks have the same capacity C.'

Example 5. $1,tw_j\,|\,1\,|\,\Delta\,|\,T$ [Savelsbergh 1986]
'In the TSPTW [traveling salesman problem with time windows] we are given in addition to the travel time $t_{i,j}$ for each pair of vertices $i,j \in V$, for each vertex i a *time window*, denoted by $[e_i,l_i]$, where e_i specifies the earliest service time and l_i the latest service time. The latter bound is strict in the sense that departing later than l_i is not allowed and causes the tour to become infeasible, whereas arriving earlier than e_i does not lead to infeasibility but merely introduces waiting time at vertex i. ... Minimize the completion time of the tour'.

Example 6. $1,choice\,|\,1\,|\,|\,T$ [Laporte and Nobert 1983]
'... we consider a generalization of the TSP in which each city is replaced by a set of cities. More precisely, we consider a city (city 0) used as the trip starting and ending point, and also n sets of cities $(S_1,S_2,\ldots,S_n)$. The problem ... consists of finding the shortest route through city 0 and at least one city taken from each S_k. As in the TSP each city may be visited only once.'

Example 7. $1,\pm\,|\,m,cap\,|\,back\,|\,\text{sum}T_i$ [Goetschalckx and Jacobs]
'The linehaul-backhaul problem is an extension of the VRP involving both delivery and pick-up points. Linehaul (delivery) points are sites which are to receive a quantity of goods from the single central DC [distribution center]. Backhaul (pick-up) points are sites which send a quantity of goods back to the DC. The quantities to be delivered and picked up are fixed and known in advance. There exists a homogeneous fleet of vehicles each of which is assumed to have a fixed capacity of some weight or volume. The crucial assumption is that all deliveries must be made before any pick-ups can be made. This is caused by the fact that the vehicles are rear-loaded and the rearrangement of the loads on the truck at delivery points is not deemed feasible. Hence, a feasible solution to the problem consists of a set of routes where all deliveries for each route are completed before any pick-ups are made and the vehicle capacity is not

violated either by the linehaul or backhaul points assigned to the route. The objective is to find such a set of routes which minimizes the total distance traveled.'

Example 8. 1,*period* | m,*cap*$_i$,*dur*$_i$ | | sumT_i [Christofides and Beasley 1984]
'In the period vehicle routing problem ... the problem is to design a set of routes for each day of a given (p-day) period. Each customer may require a number of visits by a vehicle during this period. If a customer requires k (say) visits during the period, then the visits may only occur in one of a given number of allowable k-day *combinations*.'

Example 9. 1,EDGE,*subset* | m,*dur* | / | sumC_i,sumT_i [Stern and Dror 1979]
'This paper is addressed to the problem of collecting data on household consumption of electricity for billing purposes ... Each reader has a maximum ... workshift time limit of 5 hr established by union regulations. ... Figure 2 [not reproduced here] presents a graph that corresponds to the network of streets. The heavy lined edges represent those streets that contain meters and must be covered by the meter readers while moving from house to house. Dotted edges represent streets that contain no meters but may be traversed as connecting streets if required. There are no oneway streets ... as readers proceed by foot and walking can be done on sidewalks in any direction. ... working tours may begin and end at intermediate locations of an edge. ... The primary objective is to find the minimum number of working tours needed to cover the required edges in the graph. A secondary objective, given the minimum number of tours, is to find the routes of each tour such that the total length traversed is minimal.'

Example 10. l,*tw*$_j$ | m,*cap*$_i$,*sep*,*tw*$_i$ | $\geqslant 1R/V,AV$ | sumT_i,sumP_i,sump_j [Brown and Graves 1981]
'The dispatchers, located at a central national order processing facility, must each handle several bulk terminals ... Drivers are domiciled with company-owned vehicles at the terminal ...

Delivery vehicles possess a wide variety of features relevant to their use in the dispatch. A model truck and rig ... is equipped with multiple, isolated compartments. Each compartment has a volumetric capacity specific to the density of the product contained. ...

Vehicle operating costs are specified for each proprietary truck on a customer-by-customer basis as a function of mileage and standard delivery time. Nonproprietary truck costs may also be simple functions of actual delivery time and mileage, or may be fixed point-to-point charges for each trip depending upon operating region and contract terms and duration. Each vehicle is assigned a sequence of loads for a shift with the duration of each shift determined by driver availability, vehicle availability, and contract terms. Overextension of vehicle shifts leads to overtime labor costs ...

Each order typically includes three products, usually grades of gasoline, jointly constituting a complete truck load ... and additional data regarding special equipment requirements (such as special couplings, pumps, an unmarked truck, and so forth).'

Example 11. 1,TASK | 1 | | T [Psaraftis 1983]
'In the DARP [dial-a-ride problem]'s generic version, a vehicle, initially located at point A, is called to service N customers, each of whom wishes to travel from a distinct origin to a distinct destination, and then return to A so that the total length of the route is minimized.'

Example 12. 1,$\pm$,*mw*$_j$,*subset* | m,*cap*$_i$,*tw*$_i$ | $\geqslant 1R/V,AV$ | maxT_i,sumT_i [Anthonisse, Lenstra, and Savelsbergh 1987]
'CAR is an interactive software package ... for distribution problems with the following characteristics.

- There is a single depot where several vehicles, possibly with different capacities, are stationed.
- The commodity to be transported is homogeneous in the sense that the allocation of commodities to vehicles is restricted only by the vehicle capacities.
- A vehicle can make several trips a day.
- A vehicle has a time window that specifies its availability.
- There may be both collections and deliveries. Vehicles depart from the depot with the commodities to

be delivered and eventually return to the depot with the collected commodities. Anything collected on the way is transported to the depot. Collections and deliveries may occur in any sequence on the same trip.
- An address may impose restrictions on the capabilities of the vehicle visiting it. For example, it may require special loading equipment.
- An address has one or more time windows within which service must take place.
- An address can have a priority, indicating that it must be visited.
- An address is to be visited by at most one vehicle.'

Example 13. $1,\sim,mw_j,period \mid m,cap_i,tw_i \mid /,\geqslant 1R/V,AV \mid \mathrm{sum}c_j,\mathrm{sum}T_i,\mathrm{sum}C_i$ [Bell, Dalberto, Fisher, Greenfield, Jaikumar, Kedia, Mack, and Prutzman 1983]
'The degree of freedom available to distribution management at Air Products is greater than in any other industry. They decide when to supply a customer based on the inventory level in the customer tank, how much to deliver, how to combine the different loads on a truck and how to route the truck. Thus inventory management at customer locations is integrated with vehicle scheduling and dispatching. ...

Because of the uncertainty in customer demand, ... inventory must be maintained at a specified safety-stock level. Customers are not open for delivery on every day of the week or during every hour of the day and trucks must make their deliveries within certain prescribed time windows which can vary among customers. The trucks in the fleet differ in characteristics such as capacity and operating costs. ... Finally, some trucks are incapable of serving certain customers because they are too big, require an external power source for an electric pump, and so forth. The availability of trucks, drivers and product is limited. ...

The costs that must be considered in scheduling include driver pay, tolls, and vehicle-related costs such as depreciation, fuel, and maintenance. ...

The scheduling module is used daily at each depot to produce a detailed schedule for a two- to five-day horizon, with the first day's schedule being the most important one. ... The object of the model is to maximize the value of the product delivered less the costs incurred in making these deliveries.'

Example 14. $1,\mathrm{TASK},tw_j \mid m \mid dir,full \mid \mathrm{sum}C_i,\mathrm{sum}T_i$ [Desrosiers, Soumis, and Desrochers 1984]
'A *trip* is a productive journey which may be carried out by a vehicle. The trip i is characterized by a place of origin, a destination, a duration, a cost and a time interval $[a_i,b_i]$ during which the trip must begin. An *intertrip arc* is an unproductive (i.e., empty) journey carried out by a vehicle. The intertrip arc (i,j) goes from the end of trip i to the beginning of trip j. Its duration t_{ij} and its cost c_{ij} may include respectively the duration and cost of the trip i. A *route* is a sequence of trips and intertrip arcs carried by the same vehicle. The problem is to determine routes and schedules for all the trips so as to minimize the number of vehicles and travel costs while respecting network and scheduling constraints.'

4. Conclusion

We have presented a classification scheme for a broad class of vehicle routing and scheduling problems. The majority of the models that have been considered in the literature can be classified according to this scheme. It may be a useful tool in clarifying the multitude of characteristics of practical vehicle routing and scheduling problems and in bringing some uniformity in the literature on the subject. In addition, we intend to use it as the basis for the development of a system for computer aided model construction and algorithm management in this area.

Acknowledgements

We gratefully acknowledge discussions with Jean-Marc Rousseau, Serge Roy, Francois Soumis and Leen Stougie on the design of our classification scheme.

REFERENCES

J.M. ANTHONISSE, J.K. LENSTRA, M.W.P. SAVELSBERGH (1987). *Functional Description of CAR, an Interactive System for Computer Aided Routing*, Report OS-R8716, Centre for Mathematics and Computer Science, Amsterdam.

W.J. BELL, L.M. DALBERTO, M.L. FISHER, A.J. GREENFIELD, R. JAIKUMAR, P. KEDIA, R.G. MACK, P.J. PRUTZMAN (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces 13*, 4-23.

L. BODIN, B. GOLDEN (1981). Classification in vehicle routing and scheduling. *Networks 11*, 97-108.

G.G. BROWN, G.W. GRAVES (1981). Real-time dispatch of petroleum tank trucks. *Management Sci. 27*, 19-32.

N. CHRISTOFIDES, J.E. BEASLEY (1984). The period routing problem. *Networks 14*, 237-256.

R.W. CONWAY, W.L. MAXWELL, L.W. MILLER (1967). *Theory of Scheduling*, Addison-Wesley, Reading, MA.

G.B. DANTZIG, J.H. RAMSER (1959). The truck dispatching problem. *Management Sci. 6*, 80-91.

J. DESROSIERS, F. SOUMIS, M. DESROCHERS (1984). Routing with time windows by column generation. *Networks 14*, 545-565.

M. GOETSCHALCKX, C. JACOBS (undated). *The Vehicle Routing Problem with Backhauls*, Manuscript, Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

R.L. GRAHAM, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann. Discrete Math. 5*, 287-326.

GUAN MEIGU [KWAN MEI-KO] (1962). Graphic programming using odd or even points. *Chinese Math. 1*, 273-277.

G.Y. HANDLER, P.B. MIRCHANDANI (1979). *Location on Networks: Theory and Algorithms*, MIT Press, Cambridge, MA.

W. JAQUES (1859). *The Icosian Game*, published and sold wholesale by John Jaques and Son, London.

B.J. LAGEWEG, E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN (1981). *Computer-Aided Complexity Classification of Deterministic Scheduling Problems*, Report BW 138, Mathematisch Centrum, Amsterdam.

B.J. LAGEWEG, J.K. LENSTRA, E.L. LAWLER, A.H.G. RINNOOY KAN (1982). Computer-aided complexity classification of combinatorial problems. *Comm. ACM 25*, 817-822.

G. LAPORTE, Y. NOBERT (1983). Generalized travelling salesman problem through n sets of nodes: an integer programming approach. *INFOR 21*, 61-75.

K. MENGER (1930). Das Botenproblem. K. MENGER (ed.) (1932). *Ergebnisse eines Mathematischen Kolloquiums 2*, Teubner, Leipzig, 9. Kolloquium (5.II.1930), 12.

H.N. PSARAFTIS (1983). Analysis of an O(N^2) heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem. *Transportation Res. Part B 17B*, 133-145.

D. RONEN (1987). Perspectives on practical aspects of truck routing and scheduling. *European J. Oper. Res.*, to appear.

M.W.P. SAVELSBERGH (1986). Local search for routing problems with time windows. *Ann. Oper. Res. 4*, 285-305.

H. STERN, M. DROR (1979). Routing electric meter readers. *Comput. & Oper. Res. 6*, 209-223.