

# A Heuristic Method for Analyzing Driver Scheduling Problem

Liping Zhao

**Abstract**—A heuristic approach, ZEST for ESTimator, is developed to analyze bus driver scheduling problems and produce an estimate of the number of drivers required for a bus schedule. Based on the observation that the maximum number of drivers is needed in the morning and afternoon peaks, ZEST divides the driver scheduling problem into morning and afternoon subproblems, solves each subproblem separately, and, finally, combines the solutions. The key techniques in ZEST derive from manual scheduling operations that examine the critical decision points in a bus schedule that are vital for a good driver schedule and use these decision points to develop chains of meal breaks that dovetail one driver's meal break with another driver's. ZEST can be used as a standalone estimator of driver duties or as a component of other driver scheduling approaches.

**Index Terms**—Driver scheduling problem analysis, duty estimation, heuristic method, public transport system.

## I. INTRODUCTION

THE main planning processes that support a public transport company's operation and service are the production of operational timetables, vehicle schedules, driver schedules, and rosters. These four processes are called timetabling, vehicle scheduling, driver scheduling, and rostering. Timetabling is the first activity that sets the level of service to be provided in terms of when and where vehicle journeys are to be made. Vehicle scheduling allocates these journeys to vehicles and driver scheduling assigns driver duties to cover the vehicle work. Finally, rostering combines driver duties with rest days to form work plans for drivers. Hence, driver scheduling deals with the notional rather than actual driver assignment, whereas the actual driver assignment happens in the rostering process when driver duties are combined with the rest days to form work plans for drivers. Followed by these offline planning processes are the actual operation, monitoring, or real-time scheduling and controlling of the service on a particular day.

These planning and operating processes—from timetabling to actual operations—are constrained by many complex and interdependent factors, such as network structures, perceived demand for services, vehicle availability, driver availability, labor costs, and union rules. These constraints and rules vary considerably from one company to another, particularly across national boundaries. Over the past 40 years, research efforts have been made to develop the methods and computer systems to assist these processes. Since 1975, a series of workshops

[3], [4], [6], [24], [28], [29], [32], [34] has been held to report progress and new developments in this area. Among these planning and scheduling processes, driver scheduling remains to be the most challenging problem, and new approaches have been constantly sought to solve this problem. A survey of these approaches has been given by Wren [35], [37] and Wren and Rousseau [39].

There are two general approaches to the driver scheduling problem by computer—one is based on heuristics and the other uses mathematical programming models. Heuristics solves the problem by an intuitive approach, in which the structure of the problem is interpreted, analyzed, and exploited intelligently to obtain a “reasonable” solution [19]. Heuristics was used predominantly in driver scheduling between the early 1960s and the late 1970s [7], [17], [20], [31], [33]. Although during that time, it was known that the driver scheduling problem could be formulated as an integer linear programming (ILP) problem—particularly, set covering and set partitioning [27]—no practical solution could be obtained from such a formulation until much later when computers became more powerful [18], [23]. However, mathematical programming can only solve relatively small problems because of the combinatorial nature of scheduling constraints and variables. Practical mathematical programming approaches have used heuristics to reduce the size of the problem [21], [30], and the combination has led to successful driver scheduling systems that solve large and complex problems [5], [8], [9], [11], [16], [25]. Most recently, constraint programming has been found useful in assisting mathematical programming to reduce the complexity of the scheduling problems [2], [15], [26]. A new type of heuristics, called meta-heuristics, has begun to show some promise to driver scheduling problems. This includes generic algorithms [1], [13], [14], [40], [41].

This paper describes a heuristic method for analyzing the driver scheduling problem. This method consists of three sequential steps to explore the structure of a given bus schedule. The outcome of this exploration is an estimate of the number of driver duties required for the bus schedule and a skeleton of a good driver schedule containing these duties. This method, hence called ZEST for ESTimator, derives from the manual scheduling process commonly used in the U.K. bus companies and captures experience of human schedulers. ZEST can be used to aid a mathematical approach or a standalone method to driver scheduling problems either by computer or scheduler.

ZEST derives from a decomposition approach to driver scheduling problems [36]. By 1994, it had evolved into an analytical tool [42], [44], [45], and it had been implemented on both the UNIX and personal computer (PC) platforms using the

Manuscript received November 10, 2003; revised March 26, 2004. This paper was recommended by Associate Editor Y. Pan.

The author is with the School of Informatics, University of Manchester, Manchester, M60 1QD, U.K. (e-mail: Liping.Zhao@manchester.ac.uk).

Digital Object Identifier 10.1109/TSMCA.2005.853497

C and C++ programming languages. Since then, the improvements have been made to redesign the ZEST system using new software design techniques, such as design patterns [12], [43]. As a result, a better understanding of the driver scheduling problem has been obtained. This paper is thus written with this new understanding and organized into the following sections. Section II introduces the problem of bus driver scheduling. Section III describes a crude heuristic as the foundation of ZEST, and Section IV presents the ZEST method. Section V discusses the potential applications of ZEST and compares ZEST with other approaches. Finally, Section VI presents the conclusion.

## II. PROBLEM OF BUS DRIVER SCHEDULING

The problem of bus driver scheduling is defined as follows: “Given a bus schedule that sets out the precise work to be undertaken by each of the buses in a fleet during the course of a day, it is necessary to form and allocate drivers’ duties so that every bus has a driver at all times [40].” The main objective of driver scheduling is to minimize the number of duties required and the total cost of duties. In line with this objective, a good driver schedule is the one that contains a minimum number of duties.

Bus driver scheduling is constrained by bus schedules, which define the bus work for driver duties, and labor rules, which specify a company’s policy and union agreements for driver duties. In order to understand the rest of the paper, a brief description of these scheduling constraints is given below.

A bus schedule consists of blocks of bus work. Each block represents the work of a bus from the time it leaves a parking point until its next return to a parking point. A block of bus work, or a bus for short, may consist of several pieces, each piece being the minimum indivisible work for the bus between relief opportunities. A relief opportunity is a designated timing point that defines a relief point or place at which a driver can be discharged or replaced by another driver and a relief time for such an activity to take place. An example of a partial bus schedule consisting of three buses is shown in Fig. 1, where bus 2’s relief opportunities are (0435, g), (0700, b), (0812, c), etc. A relief opportunity, e.g., (0435, g), denotes the relief time (0435) and the relief point (g). In Fig. 1, relief times are in clock time. It is important to note that a bus schedule provides all the relief opportunities, but, usually, only a subset of them is used for driver duties.

Given a bus schedule as described above, the task of driver scheduling is to assign buses to drivers to construct a driver schedule of driver duties or shifts. Furthermore, since there are usually far more relief opportunities than needed in a driver schedule, the driver scheduling problem is the problem of selecting a good subset of relief opportunities, which collectively contribute to the least number of driver duties and the lowest costs of the duties, hence, a good driver schedule. Research in the past decades has demonstrated that this is very difficult to achieve (see some ongoing research efforts in this area: [9], [10], [13]–[15], [26], and [38]). Suffice it to say, the common goal of the driver scheduling systems is to identify such a subset, irrespective of the methods they are using. Fig. 2 illustrates a driver schedule for the bus schedule given in Fig. 1.

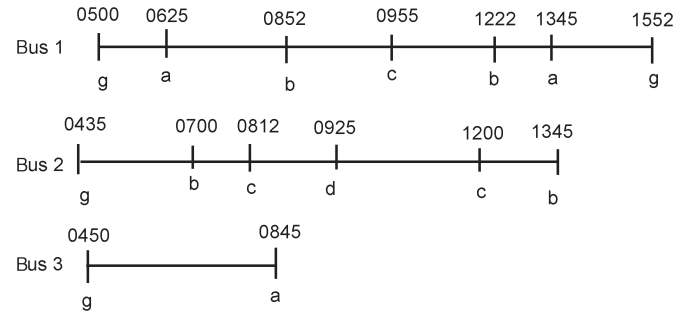


Fig. 1. Simple bus schedule showing three blocks of bus work. Each block is called a bus and has a sequence of driver relief opportunities. Each relief opportunity contains a timing point (e.g., 0935 on bus 2) and a location (e.g., the letter d on bus 2).

A driver duty is the bus work assigned to the driver who may cover several consecutive pieces of work from the same bus or several buses. Driver duties are either straight (e.g., duty 1 and duty 2 in Fig. 2) or split (e.g., duty 3 in Fig. 2). Straight duties usually correspond in duration to a normal working day for a driver, with a maximum of about 8 h in charge of a bus and a meal break of about an hour. There are three general types of straight duty, with early duties to open the bus service in the early morning and late duties to close the service in the late evening. Middle duties may be considered either as late starting early duties or as early finishing late duties, depending on its position, and are usually used to relieve early or late duty drivers. Sometimes, split duties—typically with a long spread-over up to 12.5 h and a long break in between—are required to cover both the morning and afternoon peaks. In some countries (e.g., the USA), a small number of overtime duties or trippers may be required for part-time drivers or other purposes.

In addition to the above four duty types, driver duties can also have different structures. However, in practice, two kinds of duty structure are commonly used by the bus companies, which are the two-bus duties and three-bus duties. A two-bus duty has two stretches on one or two different buses, separated by a meal break, as shown in Fig. 2. A three-bus duty also has two stretches, separated by a meal break, but one of the stretches covers spells on two different buses and the time to change buses. A duty stretch (see Fig. 2 for illustration) is a continuous period of work on one or more buses without a break (e.g., a meal break). A stretch may consist of one or more duty spells; a spell is an uninterrupted piece of work on one bus. For example, all the stretches in Fig. 2 are also called spells. Hence, when constructing a duty, one needs to consider its type and structure. There are therefore eight possibilities for consideration.

However, there is more to consider than the above constraints on the duty types and structure, as a valid duty is the one that must also meet the labor rules. As stated before, labor rules vary considerably from one company to another, particularly across national boundaries. Typically, these rules are expressed as constraints on the duration of each portion of a duty (e.g., the total working time required for a duty, the number of breaks within a duty, the meal break time, the cost of a duty). Some of these rules are hard so that they cannot be broken; some are soft to leave a certain leeway for adjustment. ZEST considers a common subset of the constraints applicable to most bus

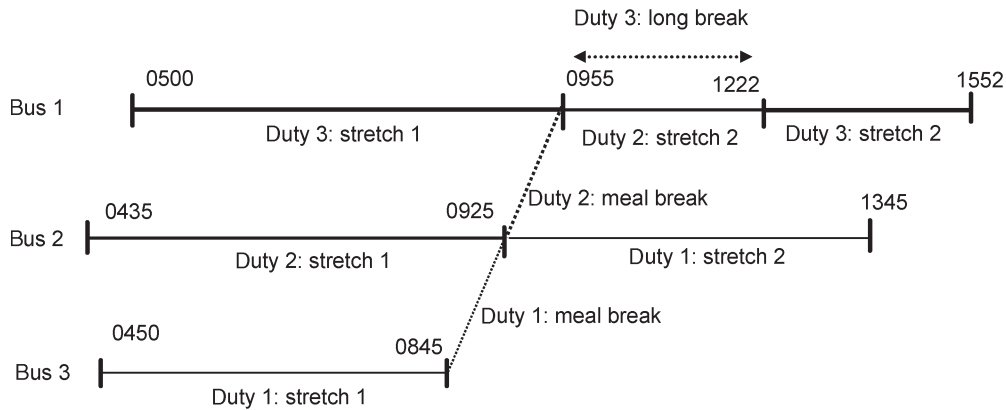


Fig. 2. Driver schedule for the bus schedule of Fig. 1. The driver schedule contains three driver duties. Each duty has two stretches and a break in-between. Duties 1 and 2 are straight duties, whereas duty 3 is a split duty that has a long break.

companies in the U.K. As the purpose of ZEST is not to produce the driver schedule, but to aid the driver scheduling process, it treats the following eight constraints as the hard constraints. The values of these constraints may vary from one company to another or from one city to another.

- 1) Duty Starting: A duty starts by signing on at the garage. There is a paid signing on allowance at the beginning of a duty, an earliest signing on time, and an earliest and a latest time for the driver to start work on a bus.
- 2) Duty Finishing: A duty finishes by signing off at the garage. There is a paid signing off allowance at the end of a duty, an earliest and a latest time for the driver to finish work on a bus, and a latest signing off time.
- 3) Travel: A driver can work on more than one bus. The time for traveling between buses is fixed.
- 4) Break Time: A duty may contain a meal break or a long break (in the case of a split duty). There is a period of time within which a meal break can start, a period within which a meal break can finish, and a maximum duration for a meal break. A meal break, including traveling time to and from the canteen, must be of at least a certain minimum duration. Meal breaks may or may not be paid.
- 5) Working Time No Break: The total working time of a duty is specified. The duration of duty stretch and spell is specified.
- 6) Duty Total Length: There is a specified length or spread-over for a duty (the time from the duty signing on to signing off).
- 7) Costs: There is a minimum and a maximum cost of a duty.
- 8) Duty Composition: These define the constraints on a valid duty. For example, they specify whether the meal break should include the traveling time to and from a canteen and whether the duty working time and the duty cost should encompass signing on and off allowances and travel time.

While the above constraints are already complex, the combinatorial nature of driver scheduling makes the problem mathematically intractable, as there are numerous ways to cover the same piece of bus work, which potentially results in a very large number of valid duties. Therefore, before a mathematical

method can be used, the number of pieces of bus work needs to be reduced by merging several pieces into one and removing relief opportunities between them. This reduction process is carried out by a heuristic method either manually or with a computer. The primary objective of heuristics in this regard is to identify the critical relief opportunities that must be retained in the bus schedule and the noncritical relief opportunities that can be removed from the bus schedule, thus reducing the number of pieces of bus work for the mathematical process. Critical relief opportunities are critical because their presence or absence can affect the quality of a driver schedule. A good driver schedule contains critical relief opportunities. This objective is in line with the objective of selecting a good subset of relief opportunities mentioned early in this section. After the application of a heuristic, the objective of a mathematical method is to remove the over-cover of each piece of bus work so that each piece of work is exactly covered by one and only one duty [8], [25].

### III. CRUDE HEURISTIC AS FOUNDATION OF THE ZEST METHOD

A normal weekday bus schedule in the U.K. consists of a morning and an afternoon peak that reflects the demand by passengers (see Fig. 3 as an example). The start of the peak period is defined as the time the maximum number of buses in service is first attained and the end of the peak period is the time this number is last attained. For example, in the bus schedule given in Fig. 3, the morning peak starts at 0810 when bus 15 pulls out of the garage and finishes at 0900 when bus 602 returns back to the garage; the afternoon peak starts at 1515 when bus 651 pulls out of the garage and finishes at 1815 when buses 651 and 15 return to the garage. The peak is a period of time rather than a specific time. The profile of a bus schedule corresponds to the variation in the demand for drivers, with more drivers required during the peaks than off peaks. However, in the context of driver scheduling, peaks are a relative notion; as the bus work is being covered and removed from the bus schedule, new peaks emerge from the outstanding bus work (called the residual bus schedule). However, for the simplicity, the reader can assume that peaks are fixed in the original bus schedule.

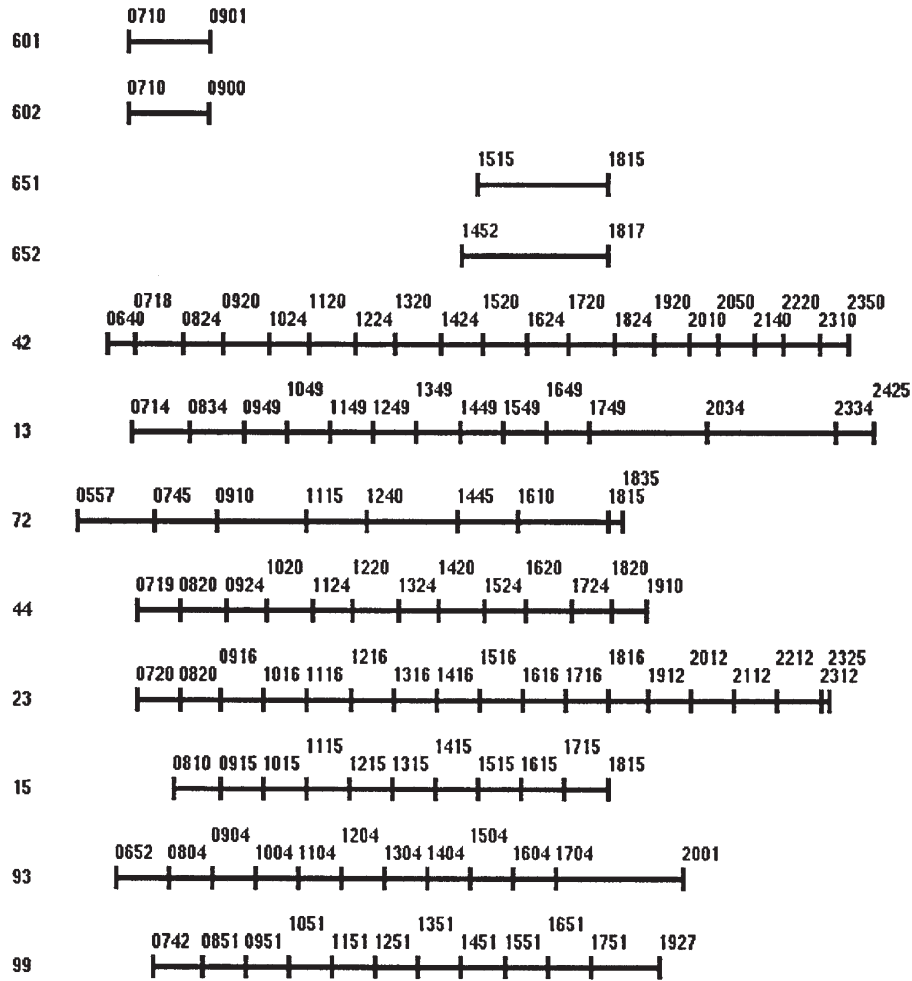


Fig. 3. Peaked bus schedule with driver relief times (relief points are omitted for brevity).

Given a peaked bus schedule as in Fig. 3, we can produce the following analysis. First, the number of buses in operation during the peak hours indicates a need for at least this number of simultaneous drivers over the peak period. Hence, it is reasonable to assume that the way to achieve the overall minimum number of driver duties for a bus schedule is to minimize the number of duties during the peaks by keeping drivers working through the peaks without breaks for as long as possible. By using this crude analysis, we can obtain a rough estimate of the number of driver duties required for a peaked bus schedule, which is the number of the morning peak buses plus the number of the afternoon peak. Thus, the bus schedule in Fig. 3 would need 20 driver duties. This crude analysis needs to be refined to be accurate, but it is a starting point.

In addition, we observe that the morning peak and the afternoon peak are not symmetric. Whereas the morning peak is shorter, building up and falling off rapidly, the afternoon peak is longer, building up gradually with school buses and then dropping off rather slowly to a lower level than after the morning peak. This observation suggests that the morning and afternoon peaks be treated differently. Hence, an intuitive approach is to divide the bus schedule into a morning schedule and an afternoon schedule, solve these two subproblems separately, and then combine the solutions of the subproblems.

The above simple, intuitive analysis provides the theoretical basis for the ZEST method.

#### IV. ZEST METHOD

Based on the above analysis, ZEST solves the morning and afternoon driver scheduling problems separately and then combines the morning and afternoon solutions into the final solution. The three ZEST solution components and their relationships are illustrated in Fig. 4, where arrows denote the sequences between the components. For example, the arrow from “Solving morning problem” to “Solving afternoon problem” means solving the morning problem first before solving the afternoon problem. However, we can also reverse the process by solving the afternoon problem first before solving the morning problem.

In line with the objective of driver scheduling, ZEST aims at producing an estimate of the minimal number of driver duties required for the bus schedule. It encapsulates techniques in an early heuristic, TRACS [17], [20], to examine the critical parts of a bus schedule and find driver duties that may resolve these difficulties. These techniques and the three ZEST solution components are described below, where the bus schedule in Fig. 3 is used as a running example.

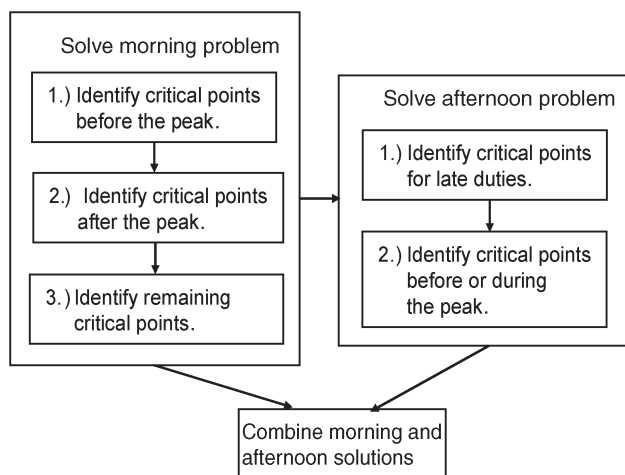


Fig. 4. ZEST solution components and their dependencies.

### A. Solving Morning Peak Problem

To solve the morning peak problem, the objective is to minimize the number of duties required for the morning peak. This is achieved by avoiding changing drivers during the peak. We first investigate how long the first driver who takes the bus out of the garage in the morning can work without a meal break. We use the duty constraints defined in Section II to determine the type of duty this driver is on and identify the latest permissible relief opportunity to release this driver for a meal break; we record the time associated with this latest relief opportunity as the marked time on bus [17]. For example, in Fig. 3, the first driver of bus 72 is assigned to an early duty because of the early duty starting constraints. Assume that the maximum permitted stretch is 4.5 h (four and a half). The marked time on this bus is therefore at 0910—we also say that the bus is marked at 0910. Peak-only buses are usually marked at the end of the bus work. Hence, bus 601 is marked at 0901 and bus 602 marked at 0900.

On the basis of marked times, buses can be classified into two categories—one with marked times before the end of the morning peak and the other with marked times after the end of the morning peak. Each bus in the first category indicates the need for a new driver to cover the bus work over the morning peak. This new driver is considered as an extra one by ZEST, because, ideally, we want the number of drivers to be the same as the number of buses during the peak. The additional drivers used to cover the morning peak are usually assigned to the middle duties. Each bus in the second category indicates the need for a new driver to cover the bus work after the morning peak. This new driver is also considered as an extra one by ZEST, for the same reason as the buses in the first category.

Therefore, the marked time on a bus serves two purposes. First, it marks the position of a meal break for the driver in relation to the position of the morning peak period. Second, it indicates the critical position where an extra duty might be introduced. Solving the morning peak problem involves 1) using the marked times as markers to find a set of critical relief opportunities and 2) using the critical relief opportunities to form meal break chains. A meal break chain is a sequence of

meal breaks that dovetail one another such that the driver who has finished the meal break can relieve another driver, who, after a meal break, can relieve the third driver, and so on. By chaining meal breaks, we can use the existing drivers to relieve each other and therefore avoid using extra drivers unnecessarily. A good driver schedule is usually associated with long meal break chains [15], because it is usually more efficient to relieve drivers who are due for a break with drivers who have already finished their meal breaks. ZEST takes the following three sequential steps to identify the critical relief opportunities in the morning bus schedule and use them to achieve the minimum number of duties.

*Step 1. Identify the critical relief opportunities before the end of the morning peak.*

If all the buses are marked after the end of the morning peak, then skip this step. All the buses in Fig. 3 are marked after the end of the morning peak and will therefore not be considered in this step.

If a bus is marked before the end of the peak, it means that its driver needs a meal break before the end of the peak. In this case, it is usually better to relieve the driver before the start of the peak so that the driver can cover the whole of the peak after the meal break. Hence, the objective in this step is to relieve the driver on a morning bus as early as possible. In doing so, both of the following two conditions must be met.

- 1) After the meal break, the driver must be able to start another bus at its beginning and then work through the whole of the morning peak to complete a valid early duty.
- 2) This driver's replacement must be able to cover the whole of the peak without violating the duty constraints.

The relief opportunity that satisfies the above two conditions is called the critical relief opportunity on the bus. If either of the above conditions is not met, then the concerned bus will be considered in step 3.

To illustrate this step, we use a simple example in Fig. 5. Assume that the morning peak is from 0759 to 0950, the maximum permitted stretch for early duties is 4.5 h and the meal break is 30 min. Bus 45 is marked at 0825 and bus 50 is marked at 1111. A crude analysis indicates that we need two duties for these two morning buses. However, since bus 45 is marked before the end of the morning peak, we will need to start an extra duty (i.e., a middle duty) at the marked time of 0825 if we relieve the driver on bus 45 at this time. If we do so, we will end up with using the three driver duties for these two buses. But instead, we relieve the driver on bus 45 at 0722 for a meal break and then put this driver back to bus 50 at 0759 to cover the whole of the morning peak and then sign off the duty at 1111; the second duty that starts at 0722 on bus 45 can also cover the whole of the peak and be relieved at 1130. By doing so, we only need two driver duties to complete the peak work on buses 45 and 50, therefore avoiding the use of the third driver for these two buses (i.e., the one who might have started at the marked time 0825 on bus 45). The relief opportunity at 0722 on bus 45 is noted as the critical relief opportunity on the bus because of its critical role in saving an extra duty. In addition, the starting times on bus 45 and bus 50 are also recorded as critical, because they are the beginnings of the duty

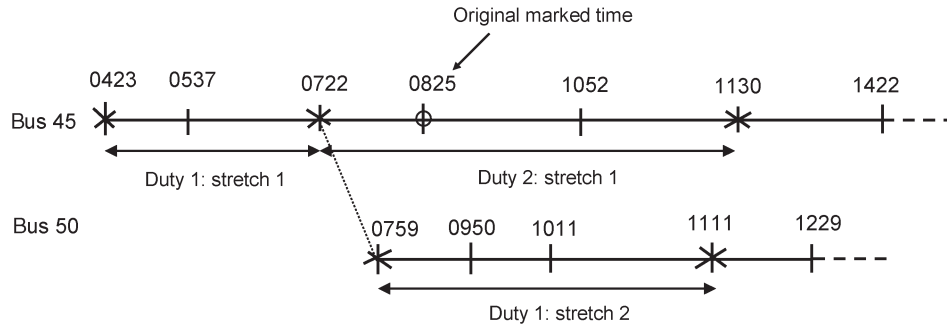


Fig. 5. Critical relief opportunities identified before the end of morning peak.

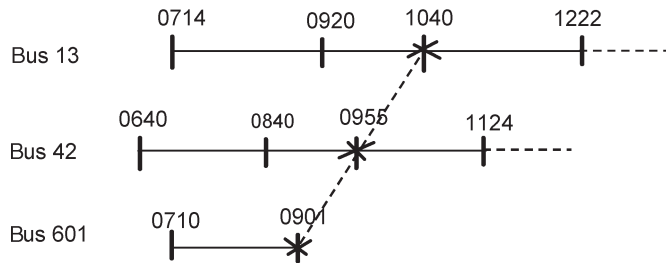


Fig. 6. Meal break chain with two breaks found on three buses. Mean break 1 starts at 0901 on bus 601 and finishes at 0955 on bus 42; mean break 2 starts at 0955 on bus 42 and finishes at 1040 on bus 13. Buses 13, 42, and 601 are from Fig. 3.

work. The marked time on bus 45 is now moved to 1130, and this bus will be considered again in step 2; bus 50 can be treated as a new bus starting at 1111, which will also be considered in step 2.

*Step 2. Identify the critical relief opportunities after the end of the morning peak.*

The objective of this step is to develop meal break chains after the end of the morning peak. All the driver duties involved in meal break chains must conform to early duty constraints, meaning that the only way we can construct meal break chains is through the construction of valid full or partial driver duties. The relief opportunities used to form meal break chains and those for starting or ending duties are critical.

A meal break chain is usually started on a peak-only bus. For example, in Fig. 3, bus 601 is a peak-only bus whose driver can relieve bus 42’s driver for a meal break at 0955; bus 42’s driver, in turn, can relieve bus 13’s driver for a meal break at 1040. These two meal breaks form a chain to save two extra drivers, who may otherwise have been used to relieve the drivers of bus 42 and bus 13. This meal break chain is shown in Fig. 6. It is worth noting that if necessary, meal break chains should also be tried in step 1 before the beginning of the morning peak.

*Step 3. Identify the remaining critical relief opportunities in the morning.*

In both steps 1 and 2, the objective is to use the existing drivers to cover each other’s meal break to form meal break chains. In step 3, the objective is to use the least number of new drivers to relieve the existing drivers for meal breaks. In doing so, we use a new driver to start a meal break chain and then let the existing drivers take over in the middle of the chain.

These new drivers are usually on middle duties. The critical relief opportunities are those used to form the meal break chains during the peak. For example, in Fig. 3, the first driver of bus 72 can be relieved at 0910 for a meal break so that the driver can take over bus 93 at 1004. A new driver on middle duty is needed to cover bus 72 from 0910. The meal break between buses 72 and 93 is illustrated in Fig. 7.

*B. Solving Afternoon Problem*

The solution of the afternoon problem is different from that of the morning problem in that the aim of the afternoon solution is principally to avoid the use of extra drivers after the end of the afternoon peak, rather than during the peak. This is because most buses will return to the garage with the decline of the peak and only a few will carry on into the late evening (e.g., Fig. 3, buses 42, 13, and 23). Nevertheless, both solutions apply the symmetric technique to identify the critical relief opportunities for the construction of meal breaks or meal break chains, in that the afternoon solution mirrors the morning solution and works backwards by first looking for critical relief opportunities for the late duties after the end of the afternoon peak and then investigating critical relief opportunities before and during the peak. There are two steps to complete the afternoon solution.

*Step 1. Identify critical relief opportunities for late duties.*

The objective in this step is to identify the minimum number of late duties required for the evening bus schedule. This step consists of the following sequences. First, we identify all the buses that can only be returned to the garage by late duties according to the duty constraints (e.g., Fig. 3 buses 42, 13, and 23). These buses are in Group A. The number of buses in Group A is used to determine the number of late duties in the driver schedule. For the remaining buses, we find all the buses that return to the garage at or after the end of the afternoon peak (e.g., Fig. 3 buses 651, 652, 72, 44, 15, 93, and 99). These buses are in Group B. We now match buses in these two groups to construct tentative late duties. The method is to link a bus from Group B to a bus in Group A with a meal break. The objective is to use one late duty to cover a late bus as well as an afternoon peak bus, hence, to avoid the use of a new duty on the late bus. For example, in Fig. 3, we can form a late duty with a meal break from 1815 on bus 15 to 1912 on bus 23. A meal break chain may be formed if the late bus that has been linked by a meal break can then be linked with another late bus. The

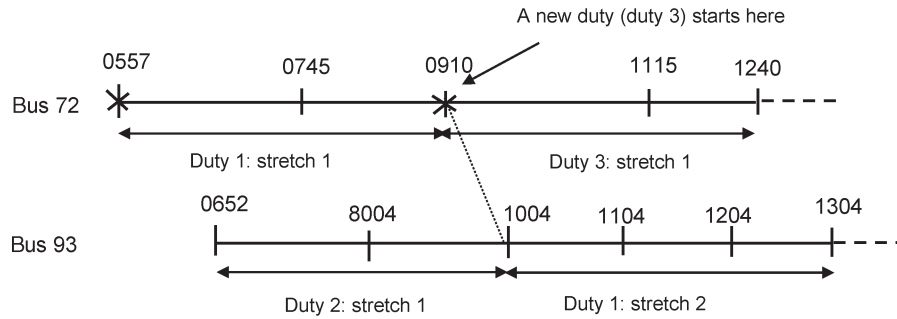


Fig. 7. Meal break from 0910 on bus 72 to 1004 on bus 93, with a new duty starting at 0910 on bus 72.

following meal break chain can thus be formed on buses 15, 23, and 42 as

$$(\text{bus } 15 : 1815) \rightarrow (\text{bus } 23 : 1912) \rightarrow (\text{bus } 42 : 2010).$$

This meal break chain involves two late duties

$$(\text{bus } 15 : 1615 - 1815) \rightarrow (\text{bus } 23 : 1912 - 2325)$$

$$(\text{bus } 23 : 1516 - 1912) \rightarrow (\text{bus } 42 : 2010 - 2350).$$

After all the late duties have been identified, the remaining bus work in the evening will be covered by middle and split duties. For example, the bus schedule in Fig. 3 requires a split or middle duty to finish at 1815 on bus 651 and a middle duty to finish at 2001 on bus 93. Drivers on these duties will usually have their meal breaks before or during the afternoon peak.

*Step 2. Identify critical relief opportunities before or during the afternoon peak.*

During this step, we investigate the following two situations.

- 1) If all the late duties can be handed over by the duties identified in the morning peak, extra duties will not be needed; otherwise, extra drivers will be used to cover the work between the morning and the late duties. The potential relief opportunities for the extra duties to finish are critical.
- 2) If the meal breaks for split and middle duty duties can be covered by the morning duties, then there is no need for extra drivers; otherwise, extra drivers will be used to cover the work for these meal breaks.

To reduce the number of extra drivers, the same technique as in the morning solution is used, that is, to use a new driver to start a meal break chain. The starting and finishing times for these meal breaks are critical. For example, in Fig. 3, two meal breaks can be formed during the afternoon peak

$$(\text{bus } 93 : 1504) \rightarrow (\text{bus } 44 : 1620)$$

$$(\text{bus } 42 : 1445) \rightarrow (\text{bus } 651 : 1515).$$

### C. Combining Solutions to Morning and Afternoon Problems

In the following, we first summarize the results obtained from solving the morning and afternoon problems of Fig. 3 bus schedule and then use the results to illustrate the process of this third ZEST component.

**Solving the Morning Peak Problem** (Section IV-A). According to the duty stretch constraints, we can obtain the

TABLE I  
EARLY DUTIES AND THEIR MEAL BREAKS

No.	Meal break	Early duty
1	(bus 601: 0901) → (bus 42: 1024)	(bus 601: 0710–0901) → (bus 42: 1024–1520)
2	(bus 42: 024) → (bus 13: 1149)	(bus 42: 0640–1024) → (bus 13: 1149–1549)
3	(bus 13: 149) → (bus 99: 1251)	(bus 13: 0714–1149) → (bus 99: 1251–1551)
4	(bus 602: 0900) → (bus 23: 1016)	(bus 602: 0710–0900) → (bus 23: 1016–1516)
5	(bus 23: 1016) → (bus 44: 1124)	(bus 23: 0720–1016) → (bus 44: 1124–1620)
6	(bus 44: 1124) → (bus 15: 1215)	(bus 44: 0719–1124) → (bus 15: 1215–1615)
7	(bus 72: 0910) → (bus 93: 1004)	(bus 72: 0557–0910) → (bus 93: 1004–1404)

marked time on each morning bus (Fig. 3) as

Bus	601	602	42	13	72
Marked time	0901	0900	1120	1149	0910
Bus	44	23	15	93	99
Marked time	1220	1216	1315	1204	1251.

The morning peak starts at 0810 when bus 15 pulls out of the garage and finishes at 0900 when bus 602 returns to the garage. Because all the morning buses are marked at the end of the morning peak, we skip step 1. From step 2, we obtain two meal break chains

$$\text{bus } 601 \rightarrow \text{bus } 42 \rightarrow \text{bus } 13 \rightarrow \text{bus } 19$$

$$\text{bus } 602 \rightarrow \text{bus } 23 \rightarrow \text{bus } 44 \rightarrow \text{bus } 15.$$

We also obtain six early duties (nos. 1–6 in Table I). From step 3, we obtain one meal break with one early duty (no. 7 in Table I) and five partial duties suitable for the first stretches of the middle and split duties (Table III: partial duties found in the morning). In summary, for the morning schedule, we have obtained

- 1) seven full early duties (Table I);
- 2) five partial duties (Table III);
- 3) a set of critical relief opportunities that are used in those full or partial duties.

**Solving the afternoon peak problem** (Section IV-B). By applying step 1, we obtain a chain of two meal breaks (bus 15 → bus 23 → bus 42) for two late duties (nos. 1–2 in

TABLE II  
LATE DUTIES AND THEIR MEAL BREAKS

No.	Meal break	Late duty
1	(bus 15: 1815) → (bus 23: 1912)	(bus 15: 1615–1815) → (bus 23: 1912–2325)
2	(bus 23: 1912) → (bus 42: 2010)	(bus 23: 1516–1912) → (bus 42: 2010–2350)
3	(bus 99: 1927) → (bus 13: 2034)	(bus 99: 1551–1927) → (bus 13: 2034–2425)

TABLE III  
PARTIAL DUTIES FOUND IN THE MORNING AND IN THE AFTERNOON

Morning partial duty	Afternoon partial duty
(bus 72: 0910–1240)	(bus 44: 1124–1620)
(bus 72: 1240–1610)	(bus 652: 1452–1817)
(bus 15: 0810–1215)	(bus 72: 1445–1835)
(bus 93: 0652–1004)	(bus 93: 1504–2001)
(bus 99: 0742–1251)	(bus 42: 1520–2010)

Table II). From step 2, we obtain one meal break for one late duty (no. 3 in Table II) and the six partial duties that are the valid second stretches for middle or split duties (Table III: partial duties found in the afternoon). In summary, for the afternoon schedule, we have obtained:

- three full late duties (Table II);
- six partial duties (Table III);
- a set of critical relief opportunities that are used in those full or partial duties.

With the above analysis, we now describe the third ZEST component: combining the morning and afternoon solutions.

The total number of full and partial duties identified in the morning and afternoon schedules is:  $7 + 5 + 3 + 6 = 21$ . This number is the maximum number of estimated duties for the whole bus schedule, which defines a high bound of the estimate. In the third component, ZEST attempts to match partial duties found in the morning with those found in the afternoon to form full duties. Every match found reduces the total number of duties by one. The partial duties that cannot be matched into the full duties are counted as full duties for the purpose of the estimation. The aim of the matching is to achieve the minimum number of the estimate.

Table IV shows four full matched duties from the partial duties in Table III; three partial duties cannot be matched and are counted as three duties.

The final estimate of the duties for the bus schedule in Fig. 3 is the total number of duties presented in Tables I, II, and IV, which is:  $7 + 3 + 7 = 17$ . The types of duty are: 8 early, 3 late, 4 middle, and 2 split.

In summary, given a bus schedule and a set of driver scheduling constraints, ZEST estimates the number of driver duties required in a good schedule. In doing so, it examines the critical parts of the bus schedule and produces a skeleton of a driver schedule that may resolve the critical parts. Such a skeleton schedule contains a set of interlinked critical relief opportunities that determine duty spells and stretches, meal breaks, and meal break chains.

## V. APPLICATIONS

### A. As a Standalone Analytical Approach

ZEST was originally intended as part of a knowledge-based approach to driver scheduling [37]. In this approach, a bus schedule would be submitted to ZEST to obtain an estimate; a driver duty would then be constructed by a primitive duty generator and deleted from the bus schedule. The residential bus schedule would be submitted to ZEST to obtain a new estimate. If the new estimate is one duty less than the previous estimate, the scheduling process would be deemed to be on course; otherwise, the process would revert to the previous state and a different duty would be formed. The process would repeat until the last duty would be completed.

However, this approach was later abandoned, because such a process was very inefficient due to the need for backtracking to the previous states. Meanwhile, ZEST had shown much promise in both efficiency and accuracy. The effort was then focused on improving ZEST still further to make it a stand-alone estimator. During the improvement, a scheduling system, IMPACS [25], was used to benchmark ZEST. IMPACS is based on a set covering approach and known to produce near-optimal driver schedules; it has been installed for about 30 transport companies, including bus, tram, and light rail systems. ZEST was evaluated against IMPACS as follows. Both systems were applied to the same problem. If the number of duties estimated by ZEST was the same or one duty less than the number of duties produced by IMPACS, the estimate was considered good and both systems proceeded to the next problem. Otherwise, ZEST was modified to account for additional rules or methods. The modified ZEST was then applied to the current and previous problems. The modification continued until ZEST produced a good estimate, i.e., the same result as IMPACS or one duty less. ZEST had been applied to seven bus schedules from different bus companies in the U.K. and had consistently produced a good estimate (Table V). It should be noted that the evaluation was based only on the total number of duties produced by the two systems. The two systems did not agree on the number of duties in types, because they have used different sets of relief opportunities. Due to the modular organization of the ZEST components, ZEST was easy to modify and adapt to new situations.

The discussion so far has shown that ZEST is a valuable analytical method to driver scheduling problems, either by computer or scheduler. However, ZEST has many potential applications. First, ZEST can be extended to form a complete driver schedule, because duties identified by ZEST, either full or partial, are valid. Partial duties can be extended to full duties by looking for suitable finishing times for these duties or by allowing some over-cover between duties. In the following, we describe two other possible applications of ZEST.

### B. As a Preprocessor for Mathematical Programming

Another advantage of ZEST began to emerge during the evaluation against IMPACS: ZEST is far more efficient. For example, for a problem that requires 89 driver duties, ZEST found the estimate in less than a minute on a 386 PC, whereas



TABLE IV  
MATCHING PARTIAL DUTIES INTO FULL DUTIES

No.	Morning partial duty	Afternoon partial duty	Match found	Duty type
1	(bus 72: 0910–1240)	(bus 44: 1324–1620)	Yes	Middle
2	(bus 72: 1240–1610)	(bus 13: 1549–2034)	Yes	Middle
3	(bus 93: 0652–1004)	(bus 72: 1445–1835)	Yes	Split
4	(bus 99: 0742–1251)	(bus 652: 1452–1817)	Yes	Split
5	(bus 15: 0810–1215)	None	No	Early
6	None	(bus 42: 1520–2010)	No	Middle
7	None	(bus 93: 1504–2001)	No	Middle

TABLE V  
COMPARISON BETWEEN ZEST AND IMPACS

Problem	System	Total duty	Duty type			
			Early	Late	Middle	Split
CS	ZEST	17	8	3	4	2
	IMPACS	17	6	3	4	4
CT	ZEST	89	35	11	29	14
	IMPACS	89	33	11	29	16
CCT	ZEST	68	25	13	19	11
	IMPACS	68	21	13	23	11
GMB	ZEST	34	12	6	10	6
	IMPACS	34	11	6	9	8
GMM	ZEST	37	18	9	10	0
	IMPACS	37	18	9	10	0
RTM	ZEST	22	7	4	5	6
	IMPACS	22	5	4	5	8
TAY	ZEST	20	10	4	5	1
	IMPACS	20	6	4	2	8

It should be noted that the comparison was based only on the total number of duties produced by the two systems. The two systems did not agree on the number of duties in types, because they have used different sets of relief opportunities.

IMPACS took about 2 h. This is because IMPACS uses a set covering approach that first generates a very large number of possible valid duties and then selects a subset of them that cover all the work on a bus schedule with the minimal cost. Although IMPACS uses some heuristics to reduce the size of the driver scheduling problem before applying the mathematical programming approach, it still requires excessive computation time for finding the subset. A recent development, TRACS II [9]–[11], improved some heuristics of IMPACS by tightening rules for generating valid duties and sieving out duties that will unlikely contribute to a good schedule. Techniques of forming meal break chains for the critical parts of the bus schedule have also been incorporated in TRACS II. Whereas TRACS II has proven more efficient than IMPACS, it is still confronted by the complexity of the scheduling problem.

The complexity of an ILP, as explained in [26], depends primarily on the number of constraints and the number of variables in a problem. In a set covering formulation of driver scheduling, the number of constraints corresponds to the number of pieces of work to be covered and the number of variables is the number of potential duties generated from the pieces of work. Since each piece of work can be covered by more than one valid

duty, the number of potential duties rises exponentially with the number of pieces. As explained in Section II, the number of pieces of work is determined by the number of relief opportunities. Hence, the complexity of a set covering driver scheduling problem directly relates to the number of relief opportunities. The success of a set covering approach therefore depends on the success in reducing the number of relief opportunities. A preprocessor based on constraint programming was proposed to select relief opportunities in TRACS II [15], [26]. The idea is to represent a scheduling problem as a constraint satisfaction problem and then to use a search algorithm to select relief opportunities. The search algorithm focuses on the formation of meal break chains and also allows some random choice of relief opportunities. Results showed that using constraint programming as a preprocessor for TRACS II can greatly reduce the time to produce a schedule [26]. As an example, by tackling a problem with 69 duties in the driver schedule, TRACS II alone took 98.42 min, whereas by including the preprocessor, it took 60.27 min. Constraint programming has also been used to select relief opportunities in a set partitioning formulation of driver scheduling [2].

It was suggested [37] that ZEST be used to select a set of relief opportunities for TRACS II. Given the heuristic rules in ZEST, the number of relief opportunities should be much smaller than that currently used in TRACS II. Resources are being sought to develop ZEST further for this purpose.

### C. As a Component in Meta-Heuristics

A new type of heuristics, called meta-heuristics, has recently been developed. The main feature of these heuristics is their efficient search algorithms and the ability to find a solution in a very large solution space. Among such heuristics are genetic algorithms (GAs) that have been applied to driver scheduling problems since 1990 [40], [41]. At first, the algorithm could only solve a small problem of about 15 duties. Further development had improved the performance of the algorithm, but the results were not satisfactory [1]. Since then, a hybrid GA—a genetic algorithm with combinatorial traits (GACT)—has produced some promising results [13], [14]. Combinatorial traits are actually meal break chains; GACT adopts a heuristic technique to select relief opportunities by forming meal break chains. Therefore, both GACT and TRACS II require preselection of relief opportunities and both use the meal break chains as a selection criterion. In this respect, ZEST can be used to select relief opportunities in a GA.

In general, all the practical driver scheduling approaches combine the techniques to reduce the complexity of the problem with the solution methods, be it heuristic or mathematical. ZEST is an important component in these approaches.

## VI. CONCLUSION

ZEST derives from manual scheduling operations and adopts the following techniques commonly used in the U.K. bus companies.

- 1) Divide the driver scheduling problem into the morning and afternoon problems and solve them separately.
- 2) Find the marked time on each bus as an approximate position for relieving the driver for a meal break.
- 3) Use the marked times to identify the critical relief opportunities around morning and afternoon peaks and use them to construct meal breaks or meal break chains.

This article has shown that these techniques, particularly the formation of meal break chains, have been used in many scheduling systems to select good relief opportunities. ZEST develops these techniques into a set of heuristic rules and strategies and integrates them into one coherent framework within which driver scheduling problems can be explored. ZEST produces a strong estimate of the number of driver duties and a skeleton driver schedule with a set of interlinked critical relief opportunities. ZEST is valuable as a standalone estimator of driver duties or as a component of other approaches that require the scheduling problem to be reduced in size for a solution. Most of the scheduling concepts defined in the ZEST research have been incorporated within EU public transport models [22].

As with any scheduling methods, ZEST needs some adaptation when applying to a new problem. The adaptation may involve adding a simple rule to direct ZEST to select a particular relief opportunity or construct a particular meal break. The modular structure of the ZEST components has made adaptation easy. Although ZEST is designed to analyze normal peaked weekday bus driver scheduling problems in the U.K., it can be adapted to solve nonpeaked problems by assuming that the end of the morning peak meets the start of the afternoon peak at 1200 noon.

In conclusion, ZEST provides a clear analysis of the driver scheduling problem and the heuristics for solving that problem; it makes scheduling knowledge more understandable and accessible by researchers and practitioners both within and outside the U.K. Most importantly, ZEST intends to show the importance of heuristics that evolved from human endeavor in solving complex problems. Such an approach, either emergent or designed, will endure and sustain, even when other powerful methods exist.

## ACKNOWLEDGMENT

This paper would not have been written without the encouragement and enthusiasm of the late Prof. J. Smith of Central Queensland University, Australia. The author's special thanks go to T. Foster for his valuable comments on the early version of this paper and to her Ph.D. supervisor, Prof. A. Wren of the University of Leeds for giving her the opportunity to work on

the public transport scheduling problems. The author is most grateful to the three reviewers for their valuable comments and detailed suggestions, which have led to many important improvements to the paper. She would like to thank the Editors of SMC-A for giving her the opportunity to revise the paper.

## REFERENCES

- [1] R. Clement and A. Wren, "Greedy genetic algorithms, optimizing mutations and bus driver scheduling," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1995, pp. 213–235.
- [2] S. Curtis, B. M. Smith, and A. Wren, "Forming bus driver schedules using constraint programming," in *Proc. Int. Conf. Practical Applications Constraint Logic Programming (PACLPL)*, London, U.K., Apr. 1999, pp. 239–254.
- [3] *Computer-Aided Transit Scheduling*, J. R. Daduna, I. Branco, and J. Paixao, Eds. Berlin, Germany: Springer-Verlag, 1995.
- [4] *Computer-Aided Transit Scheduling*, J. R. Daduna and A. Wren, Eds. Berlin, Germany: Springer-Verlag, 1988.
- [5] M. Desrochers, J. Gilbert, M. Sauve, and F. Soumis, "CREW-OPT: Subproblem modeling in a column generation approach to urban crew scheduling," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1992, pp. 395–406.
- [6] *Computer-Aided Transit Scheduling*, M. Desrochers and J. M. Rousseau, Eds. Berlin, Germany: Springer-Verlag, 1992.
- [7] S. E. G. Elias, "The use of digital computers in the economic scheduling for both man and machine in public transportation," in *Special Report, Kansas State University Bulletin 49*. Manhattan, KS: Kansas State Univ., 1964.
- [8] J. C. Falkner and D. M. Ryan, "EXPRESS: Set partitioning for bus crew scheduling in Christchurch," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1992, pp. 359–378.
- [9] S. Fores, L. Proll, and A. Wren, "An improved ILP system for driver scheduling," in *Computer-Aided Transit Scheduling*, N. H. M. Wilson, Ed. Berlin, Germany: Springer-Verlag, 1999, pp. 43–62.
- [10] —, "Experiences with a flexible driver scheduler," in *Computer-Aided Scheduling of Public Transport*. Berlin, Germany: Springer-Verlag, 2001, pp. 137–152.
- [11] —, "TRACS II: A hybrid IP/heuristic driver scheduling system for public transport," *J. Oper. Res. Soc.*, vol. 53, no. 10, pp. 1093–1100, 2002.
- [12] T. Foster and L. Zhao, "TOPS: Transport object patterns," *J. Object-Oriented Program.*, vol. 10, no. 8, pp. 26–32, 1998.
- [13] R. Kwan and A. Wren, "Hybrid generic algorithms for bus driver scheduling," in *Advanced Methods in Transportation Analysis*, I. Bianco and P. Toth, Eds. Berlin, Germany: Springer-Verlag, 1996, pp. 609–619.
- [14] R. Kwan, A. Kwan, and A. Wren, "Evolutionary driver scheduling with relief chains," *Evol. Comput.*, vol. 9, no. 4, pp. 445–460, 2001.
- [15] C. Layfield, B. M. Smith, and A. Wren, "Bus relief point selection using constraint programming," School Computing, Univ. Leeds, Leeds, U.K., Res. Rep. Ser. 98.22, Oct. 1998.
- [16] R. Lessard, J.-M. Rousseau, and D. Dupuis, "Hastus I: A mathematical programming approach to the bus driver scheduling problem," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 259–268.
- [17] B. Manington and A. Wren, "A general computer method for bus crew scheduling," presented at the Workshop Automated Techniques Scheduling Vehicle Operations Urban Public Transportation Services, Palmer House Hotel, Chicago, IL, Apr. 1975.
- [18] G. Mitra and A. P. G. Welsh, "A computer based crew scheduling system using a mathematical programming approach," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 281–296.
- [19] H. Muller-Merbach, "Heuristic methods: Structures, applications, computational experience," in *Optimization Methods for Resource Allocation*, R. Cottle and J. Krarup, Eds. London, U.K.: London Univ. Press, 1974.
- [20] M. E. Parker and B. M. Smith, "Two approaches to computer crew scheduling," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 193–222.
- [21] I. Patrikalakis and D. Xerocostas, "A new decomposition scheme of the urban transport scheduling problem," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1992, pp. 407–428.
- [22] "Reference data model for public transport," in "Drive II Programme: Eurobus Project and Harpist Task Force," TC278/WC3 Draft V4.1 prEN-Vxxx.1, European Pre-standard, 1995.

- [23] D. M. Ryan and B. A. Foster, "An integer programming approach to scheduling," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 269–280.
- [24] *Computer Scheduling of Public Transport*, J. M. Rousseau, Ed. Amsterdam, The Netherlands: North Holland, 1985.
- [25] B. M. Smith, "IMPACS—A bus crew scheduling system using integer programming," *Math. Program.*, vol. 42, no. 1, pp. 181–187, 1988.
- [26] B. M. Smith, C. J. Layfield, and A. Wren, "A constraint programming pre-processor for a bus driver scheduling system," in *Constraint Programming and Large Scale Discrete Optimization*, vol. 57, E. C. Freuder and R. J. Wallace, Eds. Providence, RI: Amer. Math. Soc., 2001, pp. 131–148.
- [27] F. Shepardson, "Modeling the bus crew scheduling problem," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1985, pp. 247–262.
- [28] *Computer-Aided Scheduling of Public Transport*, S. Voß and J. R. Daduna, Eds. Berlin, Germany: Springer-Verlag, 2001.
- [29] *Computer-Aided Transit Scheduling*, N. H. M. Wilson, Ed. Berlin, Germany: Springer-Verlag, 1999.
- [30] R. E. Ward, P. A. Durant, and A. B. Hallman, "A problem decomposing approach to scheduling the drivers and crews of mass transit systems," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 297–312.
- [31] E. B. Wilhelm, "Overview of the RUCUS package driver run cutting program (RUNS)," presented at the Workshop Automated Techniques Scheduling Vehicle Operations Urban Public Transportation Services, Palmer House Hotel, Chicago, IL, Apr. 1975.
- [32] Workshop Automated Techniques Scheduling Vehicle Operations Urban Public Transportation Services, Palmer House Hotel, Chicago, IL., Apr. 1975, unpublished preprints. Available from The Scheduling Group in School of Computing, The University of Leeds, Leeds, U.K.
- [33] A. Wren, *Computers in Transport Planning and Operation*. London, U.K.: Ian Allen, 1971.
- [34] *Computer Scheduling of Public Transport*, A. Wren, Ed. Amsterdam, The Netherlands: North Holland, 1981.
- [35] A. Wren, "General review of the use of computers in scheduling buses and their crews," in *Computer Scheduling of Public Transport*. Amsterdam, The Netherlands: North Holland, 1981, pp. 3–16.
- [36] —, *Decomposition of Large Driver Scheduling Problems*, 1983, unpublished, private circulation. Available from The Scheduling Group in School of Computing, The University of Leeds, Leeds, U.K.
- [37] —, "Heuristics ancient and modern: Transport scheduling through the ages," *J. Heuristics*, vol. 4, no. 1, pp. 87–100, 1998.
- [38] A. Wren, S. Fores, A. Kwan, R. Kwan, M. Parker, and L. Proll, "A flexible system for scheduling drivers," *J. Sched.*, vol. 6, no. 5, pp. 437–455, 2003.
- [39] A. Wren and J.-M. Rousseau, "Bus driver scheduling—An overview," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1995, pp. 174–187.
- [40] A. Wren and D. O. Wren, *Genetics, Structures and Covers—An Application to Scheduling*. Leeds, U.K.: School Computing, Univ. Leeds, Sep. 1990.
- [41] —, "A genetic algorithm for public transport driver scheduling," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 101–110, 1995.
- [42] L. Zhao, "A knowledge based driver duty estimator," Ph.D. thesis, Sch. Computing, Univ. Leeds, Leeds, U.K., 1993.
- [43] L. Zhao and T. Foster, "Driver duty: A pattern for public transport systems," *J. Object-Oriented Program.*, vol. 11, no. 4, pp. 35–39, 1998.
- [44] L. Zhao, A. Wren, and R. S. K. Kwan, "Development of a driver duty estimator," *J. Oper. Res. Soc.*, vol. 46, no. 9, pp. 1102–1110, 1995.
- [45] —, "Enriching rules in a driver duty estimator," in *Computer-Aided Transit Scheduling*. Berlin, Germany: Springer-Verlag, 1995, pp. 236–247.



**Liping Zhao** received the B.Sc.(Hons) degree in computer science from the University of Chengdu Electronic Engineering, China, in 1983, and the M.Phil. and Ph.D. degrees in computer science from the University of Leeds, U.K., in 1988 and 1993, respectively.

She is a Lecturer at the School of Informatics, University of Manchester, Manchester, U.K. She currently holds two Eng.D. research grants, awarded by the U.K. Engineering and Physical Sciences Research Council (EPSRC). Her research interests include object-oriented analysis and design, software patterns, design methods, and heuristics for public transport scheduling problems. She is one of the early researchers in object technology in the U.K. and among the first contributors who introduced software engineering principles and object-oriented technology to the public transport domain in Europe. She pioneered domain patterns relating to public transport problems and systems.

Dr. Zhao is a member of the British Computer Society.