Ant Colony Optimisation for Simulated Dynamic Multi-objective Railway Junction Rescheduling

Jayne Eaton, Shengxiang Yang, *Senior Member, IEEE,* and Mario Gongora

***Abstract*—Minimising the ongoing impact of train delays has benefits to both the users of the railway system and the railway stakeholders. However, the efficient rescheduling of trains after a perturbation is a complex real-world problem. The complexity is compounded by the fact that the problem may be both dynamic and multi-objective. The aim of this research is to investigate the ability of ant colony optimisation algorithms to solve a simulated dynamic multi-objective railway rescheduling problem and, in the process, to attempt to identify the features of the algorithms that enable them to cope with a multi-objective problem that is also dynamic. Results showed that, when the changes in the problem are large and frequent, retaining the archive of non-dominated solution between changes and updating the pheromones to reflect the new environment play an important role in enabling the algorithms to perform well on this dynamic multi-objective railway rescheduling problem.**

***Index Terms*—Train rescheduling, dynamic multi-objective optimization, ant colony optimisation, rail transportation, UK railway network.**

I. Introduction

RESCHEDULING trains after a delay is a complex real-world problem. In a busy, capricious railway network, such as the UK railway network, delays are inevitable. From signal failures to broken down trains, there are a myriad number of ways for a train to fall behind schedule. In normal railway operation conflict is avoided by the careful allocation of trains to resources at specified times. A primary delay to one train may cause it to miss its scheduled time slot which may result in conflict with another train scheduled to use that same resource. To avoid conflict a train dispatcher might have to delay other trains competing for the same resources, which will propagate the delay throughout the network.

The aim of any train dispatcher faced with train disruption is to find an optimal way to reschedule the trains in order to minimise the overall impact of the delay on the network. However, this is not a simple task. A real-world rescheduling problem may be both dynamic and multi-objective. The dynamism is a consequence of the fact that the railway system is in a constant state of movement. As trains are waiting to be rescheduled at a particular network bottleneck, more trains will be arriving. These trains may have different scheduled timetables, speed profiles and physical characteristics. The arrival of these new trains will change the nature of the problem, making it a dynamic one that changes over time. The multi-objective nature of the problem is a result of the multiple demands placed upon the train dispatcher attempting to solve a rescheduling problem. They may need to simultaneously minimise several conflicting consequences of the perturbation, such as delay, timetable deviation, energy consumption and missed connections. The conflicting nature of these objectives means that increasing the quality of one objective might have a detrimental effect on the quality of another.

The aim of this paper is to investigate the application of ant colony optimisation (ACO) to a difficult dynamic multi-objective problem (DMOP); the dynamic multi-objective railway junction rescheduling problem (DM-RJRP). Unfortunately, at the present time, Network Rail do not store the data necessary to investigate such problems, therefore the junction is simulated based on a model inspired by very thorough understanding of the real world problem.

ACO has already been shown to be effective in dynamic, combinatorial scheduling problems [17], [26] and is also very suitable for adapting to multi-objective problems due to its flexibility in being able to add multiple colonies, or multiple pheromone and heuristic matrices, to address the separate objectives. A further advantage of using ACO is that its population-based nature means that multiple trade-off solutions can be generated in one run of the algorithm, in contrast classical optimisation methods may have to run the algorithm separately for each objective [7].

Rescheduling trains after a delay is a popular research area with much interesting work being carried out [14]. However, the previous work in this area has assumed that the problem is static. There has, so far, been little work in dynamic train rescheduling problems and even less in dynamic multi-objective train rescheduling problems. Our aim is to make the following contributions to the study of railway rescheduling:

1) The investigation of a railway rescheduling problem that is both dynamic and multi-objective. As previous works consider only static or multi-objective problems, they fail to take into account the dynamic and multi-objective nature of railway scheduling problems. The investigation of such a problem is a new contribution to railway rescheduling.
2) A contribution to the field of understanding how ACO algorithms can be applied to railway rescheduling DMOPs. We attempt to identify both the features of the algorithms necessary for good performance on this DMOP and also the effect of the frequency and magnitude of change on each algorithm's performance.

Manuscript received December 14, 2015; revised October 13, 2016 and December 21, 2016; accepted February 2, 2017. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/K001310/1 and by the National Natural Science Foundation of China under Grant 61673331 (Corresponding author: *Shengxiang Yang*.)

The authors are with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort University, The Gateway, Leicester LE1 9BH, U.K. (email: syang@dmu.ac.uk).

We apply several different multi-objective ACO (MOACO) algorithms to the problem; based on a population based ACO (P-ACO) [19], and on the *MAX-MIN* Ant System (MMAS) [30]. Each algorithm uses a different method of dealing with dynamic changes. It is hoped that the performance of the algorithms will give insights into the features of the algorithm necessary for good performance on this DMOP. In addition, we compare the best ACO algorithm with NSGA-II [8], a 'state-of-the-art' multi-objective algorithm, and with First Come First Served (FCFS), a heuristic often used by railway dispatchers to resolve perturbations [5].

The complex nature of this dynamic problem means that we consider only a small section of the UK Railway Network, a junction on the Birmingham to Derby train line, called the Stenson Junction. We believe that the principles discovered in this work will aid our understanding of the problem and will be applicable to larger areas of the railway network. The problem has been modelled using a simulator built in C++. The simulator evaluates the solutions produced by the algorithms in terms of two objectives: minimising timetable deviation and minimising additional energy expenditure. Minimising timetable deviation involves minimising the difference between a train's scheduled arrival time and its rescheduled arrival time, it attempts to ensure trains arrive neither too late or too early. The second objective minimises the extra energy consumed by the trains as a result of changing the order that they pass through the junction.

The rest of the paper is organised as follows. Sect. II reviews the literature related to multi-objective and dynamic train rescheduling and the use of ACO algorithms for both multi-objective and dynamic problems. Sect. III describes the problem under investigation. Sect. IV gives details of the algorithms used in this work. An experimental study carried out to investigate the ability of ACO algorithms to solve the DM-RJRP is described in Sect. V. Finally, Sect. VI concludes this paper with a discussion of the main achievements and ideas for future work.

II. Related Work

In a multi-objective problem with conflicting objectives, there is no single solution that is able to optimise all the objectives simultaneously as any improvement in one objective may result in a deterioration in a conflicting objective. Many researchers have tackled this problem by combining the objectives into a single, often weighted, objective. The purpose of the weights is to indicate the relative importance of each objective to the problem solution. This approach will result in a single solution for each run, however, its disadvantage is that the weights will have to be determined in advance using domain knowledge. In addition, this approach assumes that the relative importance of each objective does not change over time. This may not always be the case. For example, in the early morning rush hour, a train dispatcher may wish to minimise overall delays whereas in the afternoon they may wish to maintain connections for long distance travellers. A more flexible approach is to produce a set of trade-off solutions to provide the decision maker with a choice of solutions. This will allow them to make a decision as to which solution best matches their requirements at a particular moment in time.

In order to produce a set of trade-off solutions, we need a means of comparing solutions against each other. This is achieved using the concept of dominance [7]. A solution x_1 is said to dominate a solution x_2 (denoted as $x_1 \prec x_2$) if:

1) x_1 is no worse than x_2 in all objectives and
2) x_1 is strictly better than x_2 in at least one objective.

Each solution is compared with every other solution. If a solution is not dominated by any other solution, it is added to the non-dominated set of solutions, also referred to as the Pareto optimal set (POS). The points that the Pareto-optimal solutions map to, in the objective space, is known as the Pareto optimal front (POF). The POS is the set of trade-off solutions that are presented to the decision maker. The decision maker can be confident that, in this set, no solution is any better than any other solution in terms of the trade off between objective values and it is only their particular preference at that time that makes one solution better than another.

A. Multi-Objective Train Rescheduling

Most railway rescheduling research concentrates on single objective rescheduling problems, very little work has been carried out on multi-objective problems and of those works most combine the objectives into a single weighted objective incurring the disadvantages detailed in Sect. II above.

Walker *et al.* [31] employed Branch and Bound (BB) with column and constraint generation to solve a train rescheduling problem on the Wellington Metro Line in New Zealand. Their two objectives were to minimise the deviation from the existing schedule and minimise the cost increase from the adjusted crew roster. They solved the problem by combining both the objectives into a single objective function to produce a single solution. Weston *et al.* [33] considered a rescheduling problem where the two objectives were to minimise the delay and minimise the number of broken passenger connections. They combined the two objectives into a single cost function and solved the problem using a decision tree.

Schachtebeck and Schöbel [28] and Schöbel [29] considered the bi-objective problem of minimising delay and minimising the number of missed connections for a rescheduling problem based on the railway network in the region of Harz, Germany. They combined the two objectives into a single objective and used an integer programming model to produce a single solution. Dollevoet *et al.* [9] took a similar approach for Randstad railway network in the Netherlands, combining the two objectives of minimising the delays to customers and the weighted sum of all missed connections to produce a single objective function. This was solved using CPLEX and an iterative approach that repeatedly tries to improve the assignment of trains to platforms.

Huang *et al.* [20] investigated a multi-objective timetable scheduling problem on the Beijing Yizhuang subway line. They used a genetic algorithm with a binary encoding method to find the optimal headway between trains to minimise both energy consumption and passenger travel time. Again, the objectives were combined into a single, weighted, objective

function. They found that the optimised timetable produced required two extra trains but reduced the total passenger waiting time by 23.9% and reduced energy consumption by 4.9 kWh per train. Yin *et al.* [35] considered a multi-objective rescheduling problem on a metro line where their aim was to minimise passenger travel time, passenger delay and energy consumption. To solve the problem they used an algorithm based on an approximate dynamic programming (ADP) technique and weighted each of the objectives to obtain a single solution. They found that their algorithm outperformed a heuristic (HEM) where the arrival times and departure times of all affected trains were postponed. However, to find several 'trade-off' solutions they had to rerun the algorithm with different weights which increased the execution time.

So far, there has been very little work that produces a set of Pareto optimal trade-off solutions for multi-objective railway rescheduling problems. Corman *et al.* [4] considered a bi-objective problem on a section of the Dutch railway. The two objectives they considered were those of minimising train delays and maximising the number of retained passenger and rolling stock connections. Using a BB algorithm combined with one of two heuristics, named 'Add' and 'Remove', they produced a set of trade-off solutions for the decision maker.

Lejeune *et al.* [24] considered the problem of timetabling trains to minimise the two conflicting objectives of energy consumption and running time. They used the indicator-based evolutionary algorithm (IBEA) to produce a set of 'trade-off' solutions which could be used as an aid to the timetable maker when constructing a timetable.

However, none of the above rescheduling research consider that the problem may be a dynamic one. They assume that delays occur at the beginning of the problem and that no further delays occur over the duration of the scenario. In a real-world scenario with extreme perturbations, there may be further primary train delays, or more train arrivals, which will change the nature of the problem under investigation.

B. Dynamic Train Rescheduling

The dynamic nature of the railway system is rarely considered in train rescheduling research. D'Ariano *et al.* [6] discussed the fact that speed and location modifications may happen while the algorithm is computing a solution, but concluded that the fast speed of their algorithm means that this is unlikely and therefore that such real-time variation would not have an effect on their rescheduling system. As the BB algorithm used in their work appears to have no inbuilt mechanism to cope with change, using it again after a dynamic change would effectively be a restart of the algorithm and would lose information that could potentially be usefully carried over to the new environment.

Work has started in the area of dynamic train rescheduling problems. Eaton and Yang [11] modelled a dynamic rescheduling problem on the Stenson Junction located in the UK railway network (the DRJRP). They found that a P-ACO algorithm outperformed a FCFS heuristic when the changes were frequent and of high magnitude. This suggests that in severely disrupted delay scenarios computational systems may provide a much needed aid to the train dispatcher.

The problem was extended in [12]. The difference between these two papers is that the latter paper extends the problem to a larger area of the railway network, compares more algorithms and investigates the use of random immigrants and/or elite immigrants to repair the solutions after a dynamic change. It was discovered that on the high frequency, high magnitude dynamic changes, ACO algorithms with a memory outperformed ACO algorithms that have no inbuilt mechanism to cope with dynamic change. Restarting the algorithm, by the use of random immigrants, between changes had a detrimental effect on the performance of the algorithm when the changes were large and frequent. These results illustrate the positive benefit of retaining information between changes in severely disrupted delay scenarios.

C. ACO for Multi-objective and Dynamic Problems

The modification of ACO algorithms for multi-objective problems is a popular research area. Although ACO algorithms were originally designed for single objective problems, their flexibility in allowing multiple colonies [21], [22], multiple pheromone matrices and heuristic matrices [16], [19] or a combination of both [25] makes them very suitable for problems with multiple objectives. In addition, as ACO is a population based approach, the set of trade-off solutions can be found in a single run of the algorithm.

MOACO algorithms have been applied to the real-world problems of multi-objective multicast routing [27], generating flight trajectories in hazardous weather conditions [1], task scheduling for grid over optical burst switching networks [34] and time and space assembly line balancing at the Nissan plant in Spain [3]. However, they have not as yet been applied to a multi-objective problem in the railway industry.

With regards to dynamic scheduling problems, ACO has previously been applied to the dynamic travelling salesman problem (DTSP) with good results [17], [26]. The DTSP is a combinatorial optimisation problem similar to the DRJRP. In the TSP, the objective is to find the sequence of cities for a salesman to visit that minimises the distance he has to travel whereas in the DRJRP the aim is to find the sequence of trains to pass through the junction to minimise the overall delay. One issue with the DTSP is that once the ants have converged on a solution they will still follow the same pheromone trails after a dynamic change unless the trails are updated in some way to take into account the new environment. Guntsch and Middendorf [17] tackled this problem by modifying the pheromone trails after a change while Mavrovouniotis and Yang [26] maintained diversity after a change by the use of immigrant ants. However, there has been very little investigation into real-world DMOPs using ACO algorithms. The fact that they have previously shown good results for both dynamic and multi-objective problems suggests that they may also be applicable to problems that combine both dynamic and multi-objective characteristics. The question then arises as to which features of the algorithms make them best able to cope with DMOPs. This paper aims to take a first step towards answering that question.

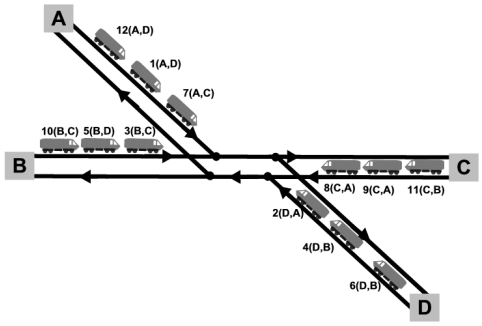


Fig. 1. The junction before a dynamic change (taken from [11]).

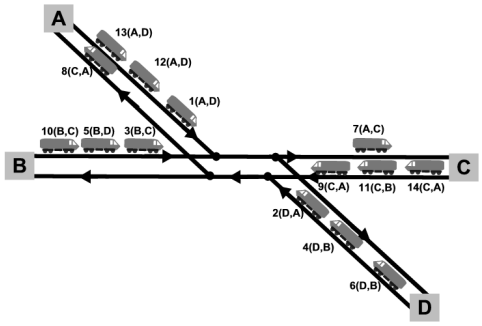


Fig. 2. The junction after a dynamic change (taken from [11]).

III. The Dynamic Multi-objective Railway Junction Rescheduling Problem (DM-RJRP)

The DM-RJRP is concerned with the sequencing of trains through two junctions on the Derby to Birmingham line. It is a microscopic model as it is modelled at the level of track block sections. The original static problem was created by Fan *et al.* [13] and has been extended in this paper to make it both dynamic and multi-objective.

The two junctions under consideration are the North Stafford and Stenson Junctions. They are both 'flat junctions' which means that the merging rail tracks require that trains cross in front of opposing trains on the same level. The junctions are connected by two sets of tracks, therefore two trains can pass through the junction at the same time as long as they are on different tracks.

The problem is a dynamic one because as trains are waiting to be rescheduled at the junction more timetabled trains will be arriving, which will change the nature of the problem. This is illustrated in Figs. 1 and 2. Fig. 1 shows the original trains waiting at a junction after a train delay. Train 1 has been delayed by 5 minutes, which means that train 7 has arrived before it on track A. However, after a while several more trains arrive at the junction, as shown in Fig. 2. Train 13 has arrived on route A while train 14 has arrived on route C. In addition, at the moment that this change occurs, trains 7 and 8 have passed through the junction and will no longer be of relevance to the problem.

A. The Problem Objectives

The DM-RJRP is not only a dynamic problem but also a multi-objective one. The two objectives under consideration are to minimise deviation from the original schedule and to minimise additional energy costs incurred by changing the order that the trains pass through the junction. These two objectives are described in more detail below.

1) Objective 1 - Minimising Timetable Deviation: When we minimise timetable deviation, we aim to minimise the difference between the train's new arrival time and its timetabled arrival time, whether that difference is positive or negative. In a rescheduling situation, trains that arrive too early can create as many problems as trains that arrive too late as both situations may result in conflict with other trains scheduled to use the same resources. We calculate the timetable deviation Dev_i of train i as in Eq. (1), where t_s is the scheduled arrival time and t_a is the actual arrival time.

$$Dev_i = |t_s - t_a| \tag{1}$$

The objective is to minimize the deviation, in minutes, for all trains at the point of change c, as shown in Eq. (2), where NT is the number of trains in the problem at change c.

$$f_1(c) = min \sum_{i=1}^{NT} Dev_i(c) \tag{2}$$

2) Objective 2 - Minimising Additional Energy Expenditure: The second objective is to minimise any extra energy expended by the trains as a result of changing the order that they pass through the junction. The energy usage calculation formulas were kindly supplied by associates at the University of Birmingham, and were taken from their microscopic railway simulator, BRaVE. We calculate the energy expended by each train on its journey from its approach station to its destination station as follows.

$$Fg = resistance + \frac{wt * gt * gd}{cos(gd)} \tag{3}$$

$$F = Fg + wt * \frac{v - u}{\Delta t} \tag{4}$$

$$E = F * d \tag{5}$$

Eq. (3) calculates the force required to overcome gravity (Fg), where wt is the weight in kilograms of the train, gt is gravity (a constant value of 9.806) and gd is gradient (zero in this case as the track is level). The resistance for the train at its current speed is found using the look-up table provided by [23]. Eq. (4) calculates the force required to move the train (F), where v is the speed (in metres per second) the train is travelling at the end of the time step and u is the speed the train was moving at the end of the previous time step. Δt is the time step which is set to 1 second. Eq. (5) calculates the energy expended (E), where (d) is the distance travelled, in metres, in the current time step. The resulting value is in joules, it is converted to kWh by dividing by 3,600,000.

The objective is to mimimise the additional energy for all trains at the point of change c, as shown in Eqs. (6) and (7), where ExE_i is the additional energy expended by train i, E_s

is the scheduled energy for train i, and E_a is the actual energy expended by train i.

$$ExE_i = argmax\{(E_s - E_a), 0.0\} \quad (6)$$

$$f_2(c) = min\sum_{i=1}^{nt} ExE_i(c) \quad (7)$$

The relationship between energy usage and train delay is complex. The original assumption made was that a slightly delayed train would use more additional energy than a seriously delayed train because the train would have had to travel faster to reduce the delay and the extra speed would use more energy. However, this was not found to be the case. Instead, a seriously delayed train was often found to use less energy. This is because, in the above equations, the amount of time a train spends waiting for the way ahead to clear before it can move again has no effect on the energy it consumes. When a train is waiting, d in Eq. (5) will be zero and consequently E will also be zero. It is recognised that, in the real world, a waiting train will use some energy, however, in the energy model represented by this set of energy equations that energy is not taken into account. A train that spends a lot of time speeding up and slowing down to avoid conflict with other trains will expend more energy than a waiting train because acceleration, especially from a standing start, uses more energy than simply travelling at a constant speed. A seriously delayed train may have spent a larger proportion of its time waiting and less time speeding up and slowing down, therefore, it will have used less energy.

Preliminary experiments found that minimising energy usage and minimising timetable deviation do conflict to some degree. The smaller the difference between the trains scheduled arrival time and its original arrival time, the more energy it is likely to expend narrowing that difference. This may be because a train that arrives very near its original scheduled time will have spent very little time waiting but will have instead performed multiple slow-downs and speed-ups, in quick succession, to maintain its schedule. The multiple speed-ups mean that it will have expended more energy on its journey. As a result of this observed conflict between these two objectives, deviation rather than delay was chosen as the objective to minimise in this study.

It is recognised that we could also have taken into account energy saved by trains when the order they pass through the junction causes them to use less than their scheduled energy. However, for simplicity, we decided to look only at extra energy expended by the trains. Future work may take into account energy saved as well as energy expended.

The aim of this problem is to find a sequence of trains to pass through the junction to minimise the objective values. As the objective values are to some degree conflicting, there will not be a single solution to the problem but a set of trade-off solutions in form of a Pareto optimal set. A characteristic of this problem is that the goal is to eventually remove the deviation and extra energy consumption from the system and to return the network to normal operation. Therefore, the aim is to end up with a single solution with an objective value of zero in each objective. This is different to many benchmark dynamic optimisation problems, where the problem constantly changes over time without ever being resolved. A further interesting feature of this problem is that it is time-linked. The decision made by the dispatcher as to which solution to choose to sequence the trains through the junction affects the trains that are available to reschedule at the next dynamic change.

TABLE I
THE SCHEDULED TIMETABLE AND ENERGY CONSUMPTION FOR EACH TRAIN

Train Number	Train Type	Route	Scheduled Arrival	Energy Consumption (kWh)
1	Class 150	A to D	12:10	23.96
2	Class 220	D to A	12:11	107.89
3	Freight	B to C	12:15	426.64
4	Class 220	D to B	12:16	63.33
5	Freight	B to D	12:16	307.15
6	Class 150	D to B	12:20	43.02
7	Freight	A to C	12:23	569.17
8	Class 150	C to A	12:21	67.90
9	Class 220	C to A	12:27	147.96
10	Class 220	B to C	12:30	140.82
11	Freight	C to B	12:39	434.57
12	Class 150	A to D	12:35	60.10

B. The Stenson Junction Train Simulator

To evaluate the performance of each algorithm, the sequence of trains in each solution has to be executed by running the trains in the specified order through the junctions. To facilitate this a simulator has been developed using C++ Visual Studio 2012. The simulator models the movement of the trains through the junction in one second time steps. The speed of the train at each second is calculated using the Improved Euler Integration, also called Heun's Method, which means that the current speed of the train is based on a combination of the train's current acceleration and an estimate of its future acceleration. Power and resistance tables supplied by Kirkwood *et al.* [23] are used to calculate a train's acceleration at time t using Newton's Second Law of Motion ($F = ma$). The tables are based on RailSys data, which is used by Network Rail as a simulation tool [32]. More details about the construction of the simulator can be found in [11].

Table I shows the type of trains used, their routes through the junction, their scheduled arrival times and their original energy consumption. The timetable was created by running all trains, in their numerical order, through the simulator and recording their arrival times. This gave a baseline measurement to be able to calculate the deviation of the trains from their original timetable after a perturbation. The trains are one of three types: Class 200 with a maximum running speed of 200km/h and a length of 187.4m: Class 150 with a maximum running speed of 120km/h and a length of 80.24m or a F2-mixed freight train with a maximum running speed of 110km/h and a length of 355m [13]. A train cannot enter a track section until the previous train has left. Therefore, the speed and length of the previous train affects how quickly a train can move into its next track section.

The simulator was made dynamic by the introduction of a specified number of trains (m) at specified time intervals (f).

m represents the magnitude of change, while f relates to the frequency of change. The new trains were chosen by repeating the timetable shown in Table I in blocks of m trains.The extra trains can be thought of as an extended timetable for the train junction and each combination of the magnitude and frequency of change was run through the simulator in order to obtain the conflict-free timetable for that dynamic scenario. A newly arrived train is not allowed to leave the station until the track section after the station is clear of all other trains. When a dynamic change occurs, any trains that have moved into the junction, or are about to move into the junction, are removed by the simulator and the remaining trains plus the additional trains are passed to the algorithm in timetable order.

C. Model Realism

At this present time, Network Rail are unable to provide the data necessary to investigate dynamic multi-objective railway rescheduling problems. Therefore, as a first step, we have developed a simulation tool that allows us to investigate such problems. To make the model as realistic as possible, we have based it on a real section of the UK railway network, simulated the mechanics of railway operation, such as interlocking and automatic fixed block technology, and used power and resistance data based on RailSys data, which is used by Network Rail as a simulation tool [32]. The model allows us to create delay scenarios with different combinations of magnitude and frequency of change which allows us to investigate the effect that the characteristics of a dynamic change has on the ability of the algorithm to provide solutions. Although such data is not stored by Network Rail at the present time, it is hoped that demonstrating the effectiveness of the algorithm in this simulated problem will provide a reason to collect and store such data in the future.

D. Model Limitations

The present model is limited by the fact that it considers only a small area of the UK railway network. Extending the model to a larger area would extend the computational complexity of the problem and could mean that in the case of the ACO algorithms more ants may be needed to obtain the same results. A further limitation of the model is that we consider only flat sections of track with no gradients. The addition of gradients into the problem would affect the energy consumption of the trains and may impact the shape of the POF obtained by the algorithms.

IV. Proposed MOACO Algorithms for the DM-RJRP

A. The Basic ACO Algorithm

ACO is an optimisation algorithm inspired by the natural world. It is based on the ability of some ant species to find the shortest path to a food source using only indirect communication in the form of pheromones [10]. Ants lay down pheromones on the ground as they move backwards and forwards from the nest to a food source. Pheromones will accumulate quicker on the shortest paths because the ants choosing those paths return faster. Ants can sense the pheromone on the ground and tend to probabilistically choose paths with the strongest pheromone concentration. A high concentration of pheromones on a trail will attract more ants which then lay down even more pheromone. In this way, the shortest path to a food source becomes marked by the strongest pheromone trail. However, once the food source is depleted retaining the high pheromone levels on this trail would be a waste of time and resources as ants would continue to follow the trail without the reward of food at the end. To prevent this from happening, pheromone trails evaporate over time and eventually disappear if they are not reinforced by the ants.

To apply this principle to an optimisation problem, it has to first be decomposed into a fully connected weighted graph $G = (V, E)$, where V is a set of vertexes or nodes and E is a set of edges or connections between the nodes. The ants move along the edges of the graph from node to node recording the nodes visited. This list of visited nodes, sometimes called the ant's tour, is one possible solution to the optimisation problem. Pheromones are deposited on the edges of the graph by the ants according to how good an ant's solution is in terms of the optimisation objective. The pheromone trails help to guide the ants to choose better nodes. Pheromones can be decreased as well as increased to model the process of evaporation which allows previous bad decisions to be forgotten. In addition to the pheromone, the edges may also be associated with a heuristic value, which is based on problem specific knowledge and provides additional guidance to the ants. An ant k, when at node i, chooses the next node j in its neighbourhood N_i^k, probabilistically as follows:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ if } j \in N_i^k, \tag{8}$$

where τ_{ij} is the pheromone information and η_{ij} is the heuristic information, α and β are constants, which determine the relative influence of the pheromone and heuristic values, respectively. An ant chooses the next node in this way with a probability of $1 - q_0$; otherwise, it chooses the next best node in terms of the pheromone and heuristic values.

In this problem each node is a train waiting to be sequenced at the junction. Unfortunately a computationally efficient and effective problem-specific heuristic is not available. Therefore, the ants rely only on the pheromone values to guide them while making their choices and the value of β is set to zero. An advantage of using only the pheromone values to guide the ants is that it reduces the amount of problem-specific knowledge needed to run the algorithm.

B. MOACOs for the DM-RJRP

There are many possible designs for MOACO algorithms as it is a popular research area and much work has been carried out on modifying ACO algorithms to make them suitable for multi-objective problems. In fact, work by López-Ibáñez and Stützle [25], where they automatically designed MOACOs for the symmetric bi-objective TSP, found that different designs produced similar quality results, suggesting that there is no single effective way to introduce a multi-objective aspect to an ACO algorithm.

In this work, two multi-objective algorithms have been chosen for investigation. The first is a multi-objective version of P-ACO developed by Guntsch and Middendorf [19]. P-ACO is a population based ACO that has an inbuilt memory. This memory allows solutions from before the change to be carried over to the new environment thus retaining information already learned by the ants between changes. The second algorithm used in this study is one designed by Alaya *et al.* [2] based on MMAS. This algorithm was chosen because its base algorithm was found to perform poorly on the DRJRP in previous work [12] and because its multi-objective modification, one colony and a pheromone matrix for each objective, is similar to that of multi-objective P-ACO. Choosing an algorithm that performed poorly allows us to investigate the modifications that are necessary to improve its performance on this DMOP. The following sections describe each of the algorithms in more detail with their dynamic adaptations.

C. Dynamic Multi-objective P-ACO

We first describe the multi-objective version of this algorithm and then the dynamic adaptation.

1) Multi-objective P-ACO: The single objective P-ACO algorithm [18] was adapted by Guntsch and Middendorf [19] to improve its performance on a multi-objective job shop scheduling problem where the two objectives to be minimised were overall tardiness and changeover costs. They modified the algorithm by adding a pheromone and heuristic matrix for each objective and by constructing the memory (P) from an archive of non-dominated solutions (Q). The memory is populated by choosing a random solution from Q plus $k-1$ closest solutions, where k is the size of the memory and the closeness of one solution to another is defined as the sum of absolute differences in objective values over all objectives. At the end of each iteration, any solutions that dominate the solutions in the archive are added to Q and any dominated solutions are removed from Q. The memory is populated from Q and the solutions in the memory are used to update each of the pheromone matrices. The ants' decision as to which node to choose next is based on a weighted summation of the separate matrices and the weights are determined using the average-weight rank method where the idea is to give an objective a higher weight if the solutions in P are better with respect to this objective compared to all solutions in Q.

In this work, we compare two different versions of dynamic multi-objective P-ACO (DM-PACO). The first makes use of the average-weight-rank method proposed by Guntsch and Middendorf to facilitate the ants' decision making. This algorithm is referred to as DM-PACO-ST. The second version of the algorithm (denoted DM-PACO-R) randomly chooses which pheromone matrix to use at each decision point. This method is the same as that used by the multi-objective version of MMAS described in Sect. IV-D1.

2) Dynamic Modification for P-ACO: This algorithm has an inbuilt memory and is able to retain information between changes. Therefore, the only modification introduced to this algorithm to allow it to cope with a dynamic change is to repair the solutions in the non-dominated archive. The repair involves removing any trains that have passed out of the problem and adding in any newly arrived trains in the order dictated by the train timetable. This is similar to the KeepElitist strategy used by Guntsch and Middendorf [17]. The pheromone values for any new trains added to the problem are initialised to τ_{min}. The repaired solutions are re-evaluated to assess their performance in the new environment and the members of the archive are reassessed for dominance: any solutions that are now dominated by any other solution in the archive are removed. The memory after a change is created from the non-dominated archive and used to reinitialise both pheromone matrices. The overall framework of this algorithm is given in Algorithm 1.

Algorithm 1 DM-PACO

```
Input P                                   ▷ The memory
Input Q                                   ▷ The non-dominated archive
Input k                                   ▷ The size of P
ConstructGraph
InitialisePheromoneTrails to τ_min
while (termination condition not satisfied) do
    ConstructSolutions
    EvaluateSolutions
    UpdateQ
    ClearP and update with k members of Q
    InitialisePheromoneTrails to τ_min
    UpdatePheromonesTrails                ▷ using solutions in P
    if change occurs then
        ReconstructGraph
        InitialisePheromoneTrails to τ_min
        RepairSolutionsInQ
        EvaluateSolutionsInQ
        UpdateQ
        ClearP and update with k members of Q
        InitialisePheromoneTrails to τ_min
        UpdatePheromonesTrails            ▷ using solutions in P
    end if
end while
```

D. Dynamic Multi-objective MMAS

The multi-objective version of MMAS we are modifying in this work is based on m-ACO$_4$(1,m), one of four algorithms designed by Alaya *et al.* for a multi-objective knapsack problem [2]. M-ACO$_4$(1,m) is similar to P-ACO in that it uses one ant colony with multiple pheromone structures, one for each objective. The following sections describe m-ACO$_4$(1,m) followed by details of the modifications made to attempt to improve its performance in a dynamic environment.

1) Multi-objective MMAS: In m-ACO$_4$(1,m), ants make their decision as to which node to choose next by randomly selecting one of the objective pheromone matrices to use in Eq. (8). At the end of an iteration, each pheromone matrix is updated separately for each objective using the best iteration ant for that objective. The update value Δ_x is based on the difference between the best-so-far ant's solution quality in objective x and the best iteration ant's solution quality in

TABLE II
FOUR DIFFERENT VERSIONS OF THE DM-MMAS ALGORITHM

	Clear Pheromones	Retain Pheromones
Clear Archive	DM-MMAS-SC	DM-MMAS-ST
Retain Archive	DM-MMAS-NC	DM-MMAS-NT

objective x as in Eq. (9), where S^x is the best solution in objective x for the current iteration and S^x_{best} is the best-so-far solution over all the iterations, including the current iteration, in objective x. The smaller the difference between the two, the larger the update.

$$\Delta_x = \frac{1}{1 + f_x S^x - f_x S^x_{best}} \tag{9}$$

As in the base MMAS algorithm, pheromone values are initialised to a maximum value. After each iteration, all pheromone trails are evaporated as in Eq. (10), where $L = E$ is the set of all pheromones and $0 < \rho \leq 1$ is the pheromone evaporation rate [10], which is a constant parameter of the algorithm. In addition, the pheromone trails are bound between a minimum τ_{min} and a maximum τ_{max} value.

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \ \forall (i,j) \in L, \tag{10}$$

Stagnation is addressed by reinitialising all trails to τ_{max} when the algorithm shows stagnation behaviour or there has been no change in the best fitness for a set number of iterations. To allow m-ACO$_4$ to produce a POS, it holds a non-dominated archive of solutions that it retains until the end of the run.

2) Dynamic Modifications for MMAS: MMAS has no in-built mechanism to cope with a dynamic change apart from the evaporation of pheromone trails, which can be slow [30]. The fact that MMAS was found to perform poorly on a single objective version of the DRJRP [12] suggests that adaptations need to be made to m-ACO$_4$(1,m) to improve its performance on the multi-objective version of the DRJRP. The goal of the modifications is to investigate the role of the pheromone trails and the archive of non-dominated solutions in the algorithm's performance. We have designed four different versions of the algorithm, summarised in Table II, that either retain the pheromones and non-dominated archive after a change or clear them. The four designs are described in more detail below:

DM-MMAS-SC: The aim with this design is to investigate how important it is to retain the pheromones after a dynamic change. For this reason, the pheromone matrix is reinitialised to τ_{max} after the change to remove all the old pheromone information. In addition, the non-dominated archive is emptied of all solutions.

DM-MMAS-ST: This version is the closest to the original behaviour of MMAS after a change in [12]. In this case, the pheromone values are retained after a change and only evaporation is used to remove old outdated decisions. As before, the non-dominated archive is emptied of all solutions.

DM-MMAS-NC: Here we want to investigate the importance of retaining the non-dominated archive between changes. Therefore, the non-dominated archive of solutions is retained after a dynamic change. However, as the archive is no longer relevant to the new environment, the solutions in it have to undergo a repair. The same repair strategy is used as for DM-PACO (see Sect. IV-C2), but, in this case the pheromone values for the new trains added to the problem are initialised to τ_{max}. In addition, the pheromone trails are cleared after each change.

DM-MMAS-NT: The purpose of this modification is to investigate both the importance of the pheromone information and the non-dominated archive after a dynamic change. Therefore, both the non-dominated archive and the pheromone information are retained after a change.

The framework of the base DM-MMAS algorithm without modifications (DM-MMAS-SC) is given in Algorithm 2.

Algorithm 2 DM-MMAS-SC

```
Input NDS                          ▷ The non-dominated archive
Input r                            ▷ Reinitialisation Interval
Input BestIterationAnt_i           ▷ in objective (i)
ConstructGraph
InitialisePheromoneTrails to τ_max
while (termination condition not satisfied) do
    ConstructSolutions
    EvaluateSolutions
    UpdateNDS ▷ with any new non-dominated solutions
    Update BestIterationAnt_i
    EvaporatePheromoneTrails
    UpdatePheromone i              ▷ using BestIterationAnt_i
    if no change in BestIterationAnt_i for r iterations then
        ReinitialisePheromone i to τ_max
    end if
    if change occurs then
        ReconstructGraph
        InitialisePheromoneTrails to τ_max
        EmptyNDS
    end if
end while
```

E. Dynamics Implementation

Solving a real-world train rescheduling problem requires consideration of how it could be implemented in a real-world railway perturbation scenario. After a delay, the trains relevant to the problem are passed to the algorithm to discover a POS of solutions. The train dispatcher then chooses the solution that best matches their objectives at that moment in time. The sequence of trains in this solution is run through the junctions until a dynamic change occurs, triggered by the arrival of more timetabled trains. At the point of change, a 'snapshot' is taken of the junctions by the simulator. The snapshot records the status of the trains, track and junction at the point of change. The newly arrived trains and the snapshot are passed to the ACO algorithm and the algorithm is run again to find a POS of solutions for the new environment. The first action the algorithm takes when it receives the updated information is to reconstruct the directed edge graph that the ants walk along to make their solutions. This is necessary since some trains will

have passed through the junction and will no longer be relevant to the problem while other trains will have been added.

As we are unable to predict the solution that a train dispatcher might select from the set of non-dominated solutions presented to them, we have to simulate this choice within the algorithm. This is achieved by randomly choosing a solution from the POS to make the snapshot of the junction at the point of change.

F. Comparison Algorithms

To compare our algorithms with other approaches for the same problem, we have repeated our experiments using NSGA-II [8], a 'state-of-the-art' multi-objective algorithm. NSGA-II is traditionally applied to continuous optimisation problems. In order to allow it to be used for this combinatorial problem where the order of trains that pass through the junction has to be feasible, we have modified the crossover and mutation operators. The purpose of crossover is to exploit previously found good solutions, Therefore, we have replaced it with an operation that, with probability p_c, performs a path-preserving local search with a parent to create a new child solution. Details of this search procedure can be found in [12]. The purpose of mutation is to explore the search space. Therefore, we have replaced it with a procedure that, with probability p_m, replaces a parent with a random feasible child solution.

In addition, we compare our approach with that of using FCFS, a heuristic often employed by train dispatchers to recover the timetable after perturbations [5]. The comparison is limited by the fact that FCFS can produce only a single solution to the problem. However, this approach is included to allow comparison with a technique used in the railway industry.

V. EXPERIMENTAL STUDY

A. Experimental Design

Algorithm parameters were established by preliminary experimentation. The best combination for DM-PACO was found to be 12 ants with a memory size of 8. Such a large memory size means that in many cases all the ants in the non-dominated set will be included in the memory and therefore the memory completely reflects the non-dominated set. A $q0$ value of 0.0 was found to perform best, which results in the ants always making a probabilistic decision about the next node to choose. For both pheromone matrices, the maximum pheromone value (τ_{max}) was set to 1, the minimum pheromone value (τ_{init}) was set to $1/n$, where n is the number of nodes, and the pheromone update value to $(\tau_{max} - \tau_{init})/k$, where k is the size of the memory. All pheromone levels were initialised to τ_{init}.

To make the algorithms more comparable, we used the same number of ants for each algorithm. The pheromone bounds for MMAS are given by $\tau_{max} = \frac{1}{C}$ and $\tau_{min} = \frac{\tau_{max}}{a}$, where C is the fitness of the best ant and a is a constant parameter of the algorithm. For both pheromone matrices, a was set to 25 and p to 0.5. As in the original MMAS algorithm [10], the $q0$ value was set to 0.0. Reinitialisation of the pheromone matrices to maximum was triggered when there had been no change in the best-so-far solution after 20 iterations.

In the case of NSGA-II, preliminary experimentation showed that the best performance was obtained with a p_c of 0.2 with a p_m of 0.2. To make it comparable with the ACO algorithms, a population size of 12 was used.

Nine different dynamic environments were investigated involving all permutations of 3 different magnitudes of change (2 trains, 5 trains, 8 trains) and 3 different frequencies of changes (5 mins, 10 mins, 15 mins). For all algorithms, the POS at the point of change was recorded. Thirty runs were completed for each algorithm on each dynamic scenario and all algorithms were run for 125 iterations before a dynamic change.

B. Performance Measures

The two goals of multi-objective optimisation are to find solutions that are as close to the POF as possible and as well spread as possible along the whole POF [7]. However, this is a difficult task in the real world when the true POF is unknown. For this reason, two different performance measures have been adopted to give insight into the performance of the algorithms. The measures are hypervolume (HV), which measures how much of the objective space is dominated by the members of the POS [37], and generational distance (GD), which measures the convergence of a solution's non-dominated set towards the POF. To calculate HV, a reference point is required. In this study, it is determined by the worst values for each objective over all algorithms and over all changes. This is to allow the values across all changes to be averaged as they all use the same reference point. To calculate GD a reference POF (POF^R) is needed. As this is a real-world problem the POF^R is unknown and therefore is created for each delay scenario from the union of all the POS for all the algorithms for that particular change. To give an offline performance measure for each run (P_r) an average was taken over all the changes, see Eq. (11), where NC is the number of changes and PM is the performance measure under consideration, either HV or GD.

$$P_r = \frac{1}{NC} \sum_{c=1}^{NC} PM_c \tag{11}$$

C. Experimental Results

The first interesting result is that there is no significant difference between DM-PACO-ST and DM-PACO-R across all the scenarios on both of the performance measures. In DM-PACO-ST, the ants base their decisions on a weighted aggregation of the pheromones for each objective using weights determined by the average-weight rank method [19], whereas in DM-PACO-R, the ants make their decision using only one, randomly selected, pheromone matrix. As there is no difference between the two versions of DM-PACO, DM-PACO-R was chosen as the comparison algorithm as it uses the same decision method as that used by DM-MMAS.

Results were tested for statistical significance using the Kruskal-Wallis test for multiple comparisons followed by the Wilcoxon rank-sum pairwise test with Bonferroni correction at a 0.05 significance level. FCFS was compared to DM-PACO-R

TABLE III
A STATISTICAL ANALYSIS OF AVERAGE HV AT 0.05 SIGNIFICANCE LEVEL

Algorithms	m=8 f=5	m=8 f=10	m=8 f=15	m=5 f=5	m=5 f=10	m=5 f=15	m=2 f=5	m=2 f=10	m=2 f=15
DM-MMAS-NT ⇔ DM-MMAS-ST	$s+$	$s+$	$\sim$	$s+$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
DM-MMAS-NT ⇔ DM-MMAS-NC	$s+$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
DM-MMAS-ST ⇔ DM-MMAS-SC	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
DM-PACO-R ⇔ DM-MMAS-NT	$\sim$	$s+$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
DM-PACO-R ⇔ NSGA-II	$s+$	$s+$	$\sim$	$s+$	$s+$	$\sim$	$\sim$	$\sim$	$\sim$
DM-PACO-R ⇔ FCFS	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$

using the one-sample Wilcoxon signed rank test as in this case we are comparing a single result with multiple results for the ACO algorithm. Table III relates to the HV performance measure. Results for the GD performance measure were similar. Therefore, for space considerations, only the HV performance table is shown here. The table shows the results of comparing Algorithm1 ⇔ Algorithm2, where the symbol "$s+$", "$s-$" or "$\sim$ " indicates that Algorithm1 is significantly better than, significantly worse than, or not significantly different from Algorithm2, respectively.

For both HV and GD, the version of m-ACO$_4$(1,m) that retains the non-dominated set and the pheromone trails after a change (DM-MMAS-NT) performs significantly better than the version that retains the pheromone trails but clears the non-dominated set after a change (DM-MMAS-ST). This significant difference in performance is apparent on the high magnitude, high and medium frequency changes ($m = 8, f = 5$ and $m = 8, f = 10$) and the medium magnitude, high frequency change ($m = 5, f = 5$). This result suggests that, in the DM-RJRP, it is very important to retain the non-dominated archive of solutions between changes when the changes are of a high frequency and of a medium to high magnitude.

Retaining the non-dominated archive between changes can be thought of as keeping a memory of the solutions found before. The continued existence of this archive provides a set of solutions to compare any new solutions to when checking for dominance. When many trains are added in short intervals, few trains will have had the opportunity to pass through the junction before the next set of trains arrives. This results in a large number of trains in the system and a correspondingly large search space for the ants to navigate. The large search space may make it difficult for the ants to find good new solutions especially as the good solutions may now have become localised in one area of the search space due to time-linked nature of the problem. Retaining and repairing the archive means that only solutions that are better than those already found are added to the archive, which guides the algorithm in its search for better solutions.

With regards to the issue of retaining pheromone values between changes, a comparison between DM-MMAS-NT and DM-MMAS-NC shows that, on the high magnitude, high frequency change ($m = 8, f = 5$), when both the pheromone trails and the non-dominated archive are retained between changes, the algorithm performs significantly better than when the non-dominated archive is retained but the pheromone trails are cleared. These results suggest that, in the high magnitude high frequency change, retaining the pheromones between changes improves the performance of the algorithm.

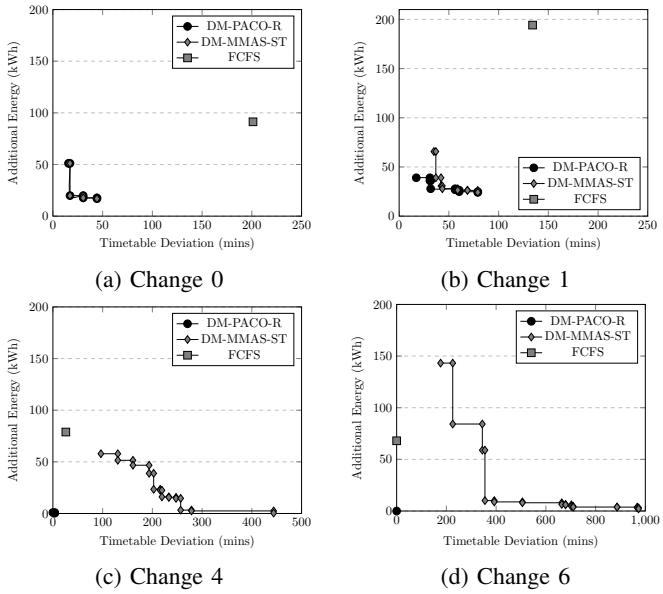
However, DM-PACO-R still significantly outperforms DM-MMAS-NT on the high magnitude, medium frequency change ($m = 8, f = 10$) even though they both retain the non-dominated archive between changes. This suggests that there may be more improvements needed for DM-MMAS to allow it to perform well in this DMOP. It is interesting that this performance difference is seen for the high magnitude, medium frequency scenario ($m = 8, f = 10$) rather than for the high magnitude, highest frequency scenario ($m = 8, f = 5$). In our previous work [12], the high magnitude, high frequency scenario showed the biggest difference in performance between the algorithms. However, an examination of the underlying results suggests that scenario $m = 8, f = 10$ is, in this DMOP, more difficult to solve than $m = 8, f = 5$. In $m = 8, f = 5$, the DM-PACO-R algorithm converged to the desired solution (0, 0) in 50% of the runs, while in $m = 8, f = 10$ it only achieved convergence in 3.33% of the runs. This suggests that $m = 8, f = 10$ is the more difficult problem to solve and also suggests that, in this DMOP, the difficulty of the problem is not only determined by the magnitude and frequency of dynamic change but also by the interaction between the objectives.

Table III shows that NSGA-II performs as well as DM-PACO-R on all the low magnitude changes ($m = 2$) and on the high and medium magnitude low frequency changes ($m = 8$, $f = 15$ and $m = 5$, $f = 15$). This suggests that the crossover and mutation operators, we used are a viable answer to the problem of how to preserve a workable order of trains to pass through the junction. However, DM-PACO-R significantly outperforms NSGA-II on the high and medium magnitude and high and medium frequency changes. This is most likely because NSGA-II has no inbuilt mechanism to cope with dynamic change and also does not retain its non-dominated archive between changes. This provides further evidence for the importance of retaining the non-dominated archive between changes.

FCFS is outperformed across all scenarios by DM-PACO-R. This is not unsurprising as FCFS produces only a single solution which may result in a lower HV score than a set of 'trade-off' solutions. For this reason, we also investigated the number of times the single solution produced by FCFS dominated the solutions in the POS produced by DM-PACO-R (Table IV). In this table, the value in square brackets shows the change number where the solution in FCFS dominated the solutions in DM-PACO-R. We can see that FCFS only dominates the solutions in DM-PACO-R when $m = 2, f = 5$ and when $m = 2, f = 15$. In each case, it was for a

TABLE IV
NUMBER OF TIMES FCFS DOMINATES THE POS PRODUCED BY DM-PACO-R IN EACH DELAY SCENARIO (SQUARE BRACKETS DENOTE THE CHANGE NUMBER)

m=8			m=5			m=2		
f=5	f=10	f=15	f=5	f=10	f=15	f=5	f=10	f=15
0	0	0	0	0	0	1[7]	0	1[3]



(a) Change 0 (b) Change 1

(c) Change 4 (d) Change 6

Fig. 3. Amalgamated POFs for DM-PACO-R and DM-MMAS-ST on m=8 f=10 for a selection of changes instances.

single change, change 7 for $m = 2, f = 5$ and change 3 for $m = 2, f = 15$. This result shows that for all the high/medium magnitude and high/medium frequency changes FCFS produces solutions that are worse than the solutions in the POS for DM-PACO-R. This is illustrated in Fig. 3, where it can be seen that the single FCFS solution is worse than the POF produced by DM-PACO-R.

Fig. 3 shows the evolution of the POF over time for the best performing algorithm (DM-PACO-R) and one of the worst performing algorithms (DM-MMAS-ST) for the delay scenario involving 8 additional trains introduced every 10 minutes. They show the POFs produced when a non-dominated set is created from the union of all the runs. The points on the front are not joined to give a smooth representation as this may be misleading for two reasons: 1). there is no guarantee that the front actually is smooth and 2). actual solutions corresponding to the intermediate vectors are unknown and may not actually exist [15]. The scale of each graph varies to make it easier to see the POFs produced.

It is apparent from Fig. 3 that the algorithms can solve the DM-RJRP to produce a POS of trade-off solutions. It is also apparent that the non-dominated fronts produced are very different for the two algorithms. Before any additional trains have been added to the problem both algorithms find a very similar POF, it is only after more trains are added that the shapes of the fronts start to diverge. After change 4, we can see a dramatic difference in the two fronts, with DM-MMAS-ST producing a large front with many solutions while DM-PACO-R has converged to a single solution with a value of zero in each objective. At a first glance, the set of graphs for DM-MMAS-ST looks the most promising as it shows a large number of non-dominated solutions on the POFs. However, paradoxically, this is not what we want in this real-world problem. In contrast, we want the effects of the delay to eventually disappear from the system to allow the trains to return to their normal running schedule. DM-PACO manages to achieve this.

It is interesting to note that overall the number of non-dominated solutions produced in this real-world problem is actually very small. This is similar to results obtained by Corman *et al.* in their work on bi-objective conflict resolution in railway traffic management [4]. In two out of three of their scenarios, they obtained an average of only 3 or 4 Pareto optimal solutions. This suggests that small numbers of Pareto optimal solutions may be a feature of real-world railway rescheduling problems.

Tables V and VI show an example set of non-dominated solutions produced for DM-PACO-R and for FCFS. Each row in Table V is a non-dominated solution with the deviation and additional energy incurred. The train order is the order the trains need to pass through the junction to give those values. We can see that FCFS has a different set of trains to sequence than DM-PACO-R. This is because, before the change occurred, different trains were sequenced and removed from the problem by FCFS than by DM-PACO-R thus resulting in a different set of trains for each algorithm to work with. This illustrates the time linked nature of the problem.

TABLE V
EXAMPLE TRADE-OFF SOLUTIONS FOR CHANGE 1 $m = 8$ AND $f = 10$ FOR DM-PACO-R FOR EACH MEMBER OF THE BEST-SO-FAR POS

Deviation (min)	Add. Energy (kWh)	Train Order
17.217	39.014	6-8-9-7-1-11-10-12-13-14-15-16-17-18-19-20
43.550	27.905	6-8-9-7-11-10-14-16-1-12-13-15-17-18-19-20
58.850	26.839	6-8-9-7-11-10-1-14-15-16-17-18-20-12-13-19
75.750	26.362	6-8-9-7-11-10-14-16-1-20-18-15-17-12-13-19
85.450	24.173	6-8-9-11-10-14-7-1-15-16-18-17-12-13-19-20

TABLE VI
FCFS SOLUTION FOR CHANGE 1 $m = 8$ AND $f = 10$

Deviation (min)	Add. Energy (kWh)	Train Order
134.167	194.201	8-12-6-3-9-5-14-10-16-13-11-15-18-17-19-20

D. Algorithm Computation Times

The experiments were run on a 2.9GHz Intel Xeon E5-2666 v3 (Haswell) processor. Table VII shows the average execution times for dynamic scenarios $m = 2, f = 15$ and $m = 8, f = 5$. These two scenarios were chosen as they are the extremes of the delay scenarios. The timing results for all algorithms were similar, therefore only the result for DM-PACO-R are shown.

TABLE VII
AVERAGE ALGORITHM EXECUTION TIMES IN MINUTES FOR DM-PACO-R ON SCENARIO $m = 2$, $f = 15$ AND $m = 8$, $f = 5$

Change	1	2	3	4	12
$m = 2, f = 15$	0.40	0.22	0.21	0.25	-
$m = 8, f = 5$	1.23	1.82	2.49	3.20	11.50

Over all changes, when two trains are added every 15 m ($m = 2, f = 15$), the average execution time is less than a minute. However, when eight trains are added every five minutes ($m = 8, f = 5$), the large number of trains in the problem increases the work of the simulator and results in an average execution time of 11.50 m for change 12. This large computation time is, of course, unacceptable in a real-world situation. However, it could be reduced by choosing a different termination condition, e.g., running the algorithm until there has been no improvement in the solutions for a predefined number of iterations. In addition, ACO is very amenable to being run in parallel [10], which would cut down the computation time considerably and make it feasible for real-time operation.

VI. CONCLUSION

The efficient rescheduling of trains after a perturbation is a complex real-world problem made more complicated by the fact that it can be both dynamic and multi-objective. It is dynamic because while trains are waiting to be rescheduled more trains will be arriving with different characteristics and schedules which will change the nature of the problem. It is multi-objective because the train dispatcher may need to consider more than one objective when making a decision as to which solution to implement. The investigation of DMOPs in the railway industry is a little un-explored area, as is the application of ACO algorithms to such problems.

An experimental study was carried out to investigate the ability of ACO algorithms to solve a simulated dynamic multi-objective rescheduling problem in the railway industry. An additional goal of this work was to attempt to identify the features of an ACO algorithm that make it suitable for coping with both the dynamic as well as multi-objective nature of this problem. The study involved the use of several multi-objective ACO algorithms. Two of the algorithms were based on multi-objective P-ACO, the others were based on different variations of a multi-objective MMAS algorithm where the aim of the modifications was to improve the performance of the algorithm on the DM-RJRP.

It is apparent that all the ACO algorithms can find a POS of solutions for the DM-RJRP. However, the algorithm based on P-ACO performs better than the algorithms based on MMAS. The performance of multi-objective MMAS can be improved, on this problem, by retaining the non-dominated archive between changes. However, for a comparable performance with DM-PACO-R, on scenarios with large and frequent changes, multi-objective MMAS also benefits from using the solution employed to make the snapshot of the junction to update the pheromone trails after a change. The best performing algoirithm DM-PACO-R also outperformed NSGA-II and FCFS.

An interesting observation in this work is that a scenario that was more difficult for the algorithm to solve in the dynamic single objective version of this problem was not necessarily the most difficult scenario to solve when the problem was made multi-objective. This suggests that the problem difficulty is not only influenced by the magnitude and frequency of dynamic change but also by the interaction between the objectives.

This work has concentrated on modifications to the algorithm after it encounters a dynamic change. It is feasible that the internal mechanisms of the algorithms may also have an effect on their ability to solve this DMOP. For example, DM-MMAS updates the pheromones with the best iteration ant in each objective while DM-PACO updates with the ants in a memory created from the non-dominated set. In addition, it is possible that NSGA-II's performance could be improved by modifications to make it able to retain information between changes such as the introduction of elite immigrants. In future work, we aim to investigate the effect of these internal mechanisms on the algorithms' performance.

The fact that the model used to explore this problem simulates the physical movement of trains through the junctions, means that on the high magnitude, high frequency changes the time taken to produce a solution is unrealistically long. In addition, this work is focused on a small area of the railway network and does not take into account the effect that changes made in a local area will have on the global behaviour of the network. For this reason, our future work will concentrate on an event-based, macroscopic model of the railway that takes into account the movements of the trains between timing points on a train's journey. This new model will allow us to extend our work to a larger area of the railway network and thus take into account the more global impact of delays. It is believed that the principles learned here can be carried over to this new model to allow us to design algorithms for larger railway rescheduling problems.

ACKNOWLEDGMENT

The authors would like to thank Dr Simon Coupland and Dr Steve Ackland at the Centre for Computational Intelligence, De Montfort University, UK for their help and advice on the physics of train simulation, and Dr Dave Kirkwood at the University of Birmingham, UK for sharing his expertise on train simulations and supplying the power and resistance tables and the energy formula used in the simulation.

REFERENCES

[1] S. Alam, L. Bui, H. Abbass, and M. Barlow, "Pareto meta-heuristics for generating safe flight trajectories under weather hazards," in *Simulated Evolution and Learning*, ser. Lecture Notes in Computer Science, vol. 4247, 2006, pp. 829–836.

[2] I. Alaya, C. Solnon, and K. Ghedira, "Ant colony optimization for multi-objective optimization problems," in *Proc. 19th IEEE Int. Conf. on Tools with Artif. Intell.*, vol. 1, 2007, pp. 450–457.

[3] M. Chica, Ó. Cordón, S. Damas, and J. Bautista, "Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search," *Inform. Sci.*, vol. 180, no. 18, pp. 3465–3487, 2010.

[4] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Bi-objective conflict detection and resolution in railway traffic management," *Transport. Res. Part C: Emerging Tech.*, vol. 20, no. 1, pp. 79–94, 2012.
[5] A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A branch and bound algorithm for scheduling trains in a railway network," *Europ. J. of Oper. Res.*, vol. 183, no. 2, pp. 643–657, 2007.
[6] A. D'Ariano, M. Pranzo, and I. Hansen, "Conflict resolution and train speed coordination for solving real-time timetable perturbations," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 2, pp. 208–222, June 2007.
[7] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.
[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[9] T. Dollevoet, D. Huisman, L. Kroon, M. Schmidt, and A. Schobel, "Delay management including capacities of stations," *Transport. Sci.*, vol. 49, no. 2, pp. 185–203, 2015.
[10] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.
[11] J. Eaton and S. Yang, "Dynamic railway junction rescheduling using population based ant colony optimisation," in *Proc. 14th UK Workshop on Comput. Intell. (UKCI 2014)*, 2014, pp. 1–8.
[12] J. Eaton, S. Yang, and M. Mavrovouniotis, "Ant colony optimization with immigrants schemes for the dynamic railway junction rescheduling problem with multiple delays," *Soft Comput.*, vol. 20, no. 8, pp. 2951–2966, Aug 2016.
[13] B. Fan, C. Roberts, and P. Weston, "A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios," *Journal of Rail Transport Planning & Management*, vol. 2, pp. 23–33, 2012.
[14] W. Fang, S. Yang, and X. Yao, "A survey on problem models and solution approaches to rescheduling in railway networks," *IEEE Trans. on Intell. Transport. Syst.*, vol. 16, no. 6, pp. 2997–3016, Dec 2015.
[15] C. Fonseca and P. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Parallel Problem Solving from Nature – PPSN IV*, ser. Lecture Notes in Computer Science, 1996, vol. 1141, pp. 584–593.
[16] C. García-Martínez, O. Cordón, and F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria tsp," *Europ. J. of Oper. Res.*, vol. 180, no. 1, pp. 116–148, 2007.
[17] M. Guntsch and M. Middendorf, "Pheromone modification strategies for ant algorithms applied to dynamic TSP," in *Appl. of Evol. Comput.*, 2001, pp. 213–222.
[18] M. Guntsch and M. Middendorf, "Applying population based ACO to dynamic optimization problems," in *Ant Algorithms*, 2002, pp. 111–122.
[19] M. Guntsch and M. Middendorf, "Solving multi-criteria optimization problems with population-based ACO," in *Evol. Multi-Criterion Optim.*, ser. Lecture Notes in Computer Science, 2003, vol. 2632, pp. 464–478.
[20] Y. Huang, L. Yang, T. Tang, F. Cao, and Z. Gao, "Saving energy and improving service quality: bicriteria train scheduling in urban rail transit systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 17, no. 12, pp. 3364–3379, Dec 2016.
[21] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Evol. Multi-Criterion Optim.*, 2001, pp. 359–372.
[22] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. on Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec 2013.
[23] D. Kirkwood, C. Roberts and F. Schmid, "Validation of railway network simulation software," in *Copenhagen: IAROR 5th Int. Conf. on Railway Operations Modelling and Analysis*, 2013.
[24] A. Lejeune, R. Chevrier, P.-O. Vandanjon, and J. Rodriguez, "Towards eco-aware timetabling: evolutionary approach and cascading initialization strategy for the bi-objective optimization of train running times," *IET Intell. Transport Syst.*, vol 10, no. 7, pp. 483–494, 2016.
[25] M. López-Ibáñez and T. Stützle, "The automatic design of multi-objective ant colony optimization algorithms," *IEEE Trans. on Evol. Comput.*, vol. 16, no. 6, pp. 861–875, Dec 2012.
[26] M. Mavrovouniotis and S. Yang, "Ant colony optimization algorithms with immigrants schemes for the dynamic travelling salesman problem," in S. Yang and X. Yao (Eds.), *Evolutionary Computation for Dynamic Optimization Problems*, ser. Studies in Computational Intelligence, vol. 490, Springer Berlin Heidelberg, 2013, pp. 317–341.
[27] D. Pinto and B. Barán, "Solving multiobjective multicast routing problem with a new ant colony optimization approach," in *Proc. 3rd Int. IFIP/ACM Latin American Conf. on Networking*, 2005, pp. 11–19.
[28] M. Schachtebeck and A. Schöbel, "IP-based techniques for delay management with priority decisions." *ATMOS*, vol. 8002, 2008.
[29] A. Schöbel, "Capacity constraints in delay management," *Public Transport*, vol. 1, no. 2, pp. 135–154, 2009.
[30] T. Stutzle and H. Hoos, "MAX-MIN ant system and local search for the traveling salesman problem," in *Proc. 1997 IEEE Int. Conf. on Evol. Comput.*, 1997, pp. 309–314.
[31] C. G. Walker, J. N. Snowdon, and D. M. Ryan, "Simultaneous disruption recovery of a train timetable and crew roster in real time," *Comput. & Oper. Res.*, vol. 32, no. 8, pp. 2077–2094, 2005.
[32] R. Watson, "Train planning in a fragmented railway - a British perspective," Ph.D. dissertation, © Robert Watson, 2008.
[33] P. Weston, C. Goodman, R. Takagi, C. Bouch, J. Armstrong, and J. Preston, "Minimising train delays in a mixed traffic railway network with consideration of passenger delay," in *Proc. 7th World Congr. on Railway Research*, 2006.
[34] Y. Yang, G. Wu, J. Chen, and W. Dai, "Multi-objective optimization based on ant colony optimization in grid over optical burst switching networks," *Expert Syst. with Appl.*, vol. 37, no. 2, pp. 1769–1775, 2010.
[35] J. Yin, T. Tang, L. Yang, Z. Gao, and B. Ran, "Energy-efficient metro train rescheduling with uncertain time-variant passenger demands: An approximate dynamic programming approach," in *Transportation Research Part B: Methodological*, vol. 91, pp. 178–210, 2016.
[36] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des., Optimisation Control*, 2002, pp. 95–100.
[37] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration," in *Evol. Multi-criterion Optim.*, 2007, pp. 862–876.

Jayne Eaton received a BSc in Computer Science from Nottingham Trent University, UK in 2006 and the MSc in Intelligent Systems and Robotics from De Montfort University, UK in 2013. She is currently working towards a PhD with De Montfort University, UK. Her current research interests are focused on using evolutionary computation algorithms for dynamic and multi-objective scheduling problems in the railway industry.

Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1993, 1996, and 1999, respectively. He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 220 publications. His current research interests include evolutionary computation, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications. He serves as an Associate Editor or Editorial Board Member of 8 international journals, such as the *IEEE Transactions on Cybernetics*, *Information Sciences*, *Evolutionary Computation*, *Neurocomputing*, and *Soft Computing*.

Mario Gongora is a Principal Lecturer in the School of Computer Science and Informatics, De Montfort University. He got his PhD from the University of Warwick (UK). He is the deputy director of the Centre for Computational Intelligence (CCI), and his research includes the application of Artificial Intelligence techniques to the identification, modelling, simulation and control of complex systems. His expertise is mainly in using evolutionary computing and biologically inspired methods for this purpose. Dr. Gongora also works in close contact with industry, applying his research results in the analysis of consumer behaviour and other complex industrial processes.