Journal of the Operational Research Society (2005) 56, 426–438

www.palgrave-journals.com/jors

Ant colony optimization for the examination scheduling problem

KA Dowsland[1] and JM Thompson[2]*

[1]*Gower Optimal Algorithms Ltd. and Nottingham University, UK; and* [2]*Cardiff University, UK*

Ant colony optimization is an evolutionary search procedure based on the way that ant colonies cooperate in locating shortest routes to food sources. Early implementations focussed on the travelling salesman and other routing problems but it is now being applied to an increasingly diverse range of combinatorial optimization problems. This paper is concerned with its application to the examination scheduling problem. It builds on an existing implementation for the graph colouring problem to produce clash-free timetables and goes on to consider the introduction of a number of additional practical constraints and objectives. A number of enhancements and modifications to the original algorithm are introduced and evaluated. Results based on real-examination scheduling problems including standard benchmark data (the Carter data set) show that the final implementation is able to compete effectively with the best-known solution approaches to the problem.

Journal of the Operational Research Society (2005) **56,** 426–438. doi:10.1057/palgrave.jors.2601830
Published online 10 November 2004

Keywords: ant algorithms; evolutionary algorithms; graph colouring; examination timetabling

Introduction

Ant algorithms are a family of evolutionary meta-heuristics whose operation is loosely based on the way that colonies of ants cooperate by laying pheromone trails. Initial applications focussed on routing and related problems, where the analogy between the biological metaphor and the optimization problem is a natural one.[1] In 1997, their scope was extended when Costa and Hertz[2] suggested a framework for an ant algorithm approach to assignment-type problems and supported their ideas by a case study based on the graph colouring problem. They use a series of experiments based on random graphs to show that their implementation of the ant algorithm, ANTCOL, can provide reasonable results. However, they also point out that the results do not match those of their own evolutionary descent method,[3] while Vesel and Žerovnik[4] show that ANTCOL is outperformed by an adaptation of an approach due to Petford and Welsh.[5] This suggests that in its current form ANTCOL is not able to compete with other approaches as an all-purpose graph colouring algorithm, but the results are sufficiently encouraging to motivate further development and experimentation. This paper is concerned with the performance of a family of algorithms based on ANTCOL for colouring graphs derived from the practical problem of examination scheduling. As pointed out by Carter[6] the structure of graphs derived from examination scheduling problems often differs from that of randomly generated graphs, and heuristic performance may not be consistent between the two groups of problems. We therefore start by carrying out experiments similar to those of Costa and Hertz[2] on graphs derived from examination scheduling data available in the public domain. We then go on to develop a series of improvements based on the additional information that is usually available in a practical examination scheduling environment. These include changes to the evaluation function and the introduction of an intelligent diversification strategy. The focus of the work described in this paper is limited to finding feasible timetables (ie timetables with no clashes) but our ultimate aim is to develop an ant algorithm that is able to deal with conflicting constraints and secondary objectives, such as time-windows, seating capacities and back-to-back exams. We therefore comment on our results in this context. Some discussion as to how the proposed improvements could be adapted for use in other ant colony optimization implementations is also included.

Our decision to experiment with ant algorithms in the solution of examination scheduling problems was motivated by several factors. First, although the problem has been widely researched, there is still no definitive solution approach that is able to provide excellent solutions across the broad spectrum of problem instances. Secondly, it is well known that the natural model for the problem is that of graph colouring, and although the results produced in Vesel and Žerovnik[4] suggest that there may be better algorithms for pure graph colouring problems, the results are sufficiently good to encourage further research. Thirdly, one of the basic ingredients of the ant algorithm is a randomized greedy construction heuristic in which each option is selected

**Correspondence: JM Thompson, School of Mathematics, Cardiff University, PO Box 926, Cardiff CF24 4YH, Wales, UK.*
E-mail: ThompsonJM1@cardiff.ac.uk

with a probability proportional to its perceived quality. As the algorithm progresses the probabilities are altered to encourage the selection of elements that have featured in good solutions. Thus unlike many other meta-heuristics which construct one or more initial solutions, and then spend most of the search effort moving away from these, ant algorithms repeatedly construct new solutions from scratch throughout the duration of the search. The process can therefore be regarded as that of tuning a simple constructive approach. This is relevant from an examination scheduling point-of-view, as there has been considerable success based on such approaches. Although some of the early implementations using greedy construction alone cannot compete with today's state of the art approaches, many recent implementations combine a greedy approach with other heuristics to provide improved performance. For example, Ross *et al*[7] use a genetic algorithm in combination with randomized greedy construction, and Carter *et al*[8] describe a greedy algorithm with limited backtracking.

The remainder of this paper is organized as follows. In the next section, the problem and the graph colouring model that underlies it are defined. This is followed by a summary of the ant algorithm and the way in which it is implemented by Costa and Hertz[2] for the graph colouring problem. We then outline some potential problems of this implementation and go on to describe the modifications introduced to overcome these. Empirical evidence comparing the performance of these modifications is followed by a discussion on the relevance of these results in the context of extending the algorithm to include secondary objectives and/or soft constraints. Suggestions for generalizing the ideas introduced here to a wider class of problems are also given.

The problem

The examination scheduling problem is that of allocating a set of exams to a given set of timeslots so that no candidate is scheduled to sit more than one exam at the same time. The precise definition differs from institution to institution in terms of both the additional hard, or binding, constraints, and the soft constraints or secondary objectives. Examples of hard constraints include an upper bound on the numbers of candidates who can sit in any one session due to room capacities and time-window constraints in which certain exams cannot take place in certain sessions (including exams that have been pre-assigned to a particular session). Soft constraints or secondary objectives usually include some measure of spreading the exams for individual students. Examples include minimizing the number of occurrences of a student having exams in consecutive slots, two or more exams on the same day, k exams in a row, k exams in n periods, etc. Many institutions also request that large exams are scheduled early to facilitate marking.

The basic problem of scheduling all the exams without clashes has a natural model as a graph colouring problem, in which the vertices are defined by the exams and two vertices are adjacent if the exams they represent have a student in common. If k timeslots are available then any feasible colouring (ie an allocation of colours to the vertices such that adjacent vertices have different colours) in k or less colours will represent a feasible timetable in which all vertices in the same colour are allocated to the same timeslot. As detailed by Balakrishnan,[9] time-window constraints can also be incorporated into this model as follows. A set of k vertices, each representing a timeslot, are added to the graph. Each timeslot vertex is adjacent to all other timeslot vertices, thus each will be coloured differently. An edge is also added between each timeslot vertex and any exams that cannot take place within that timeslot. Any feasible colouring of this graph will then respect the time-window constraints.

The other common constraints cannot be added directly to this model, although it is easy to include the room capacity constraints into any construction approach to the graph-colouring algorithm, simply by disallowing the addition of any vertex to any colour class where the upper bound on the number of candidates would be exceeded. As mentioned above our ultimate aim is to produce an ant algorithm that is able to handle all the above constraints. It is therefore important that our graph-colouring implementation must demonstrate a strong ability to seek out optimal colourings as this will be somewhat diluted by the introduction of conflicting objectives and constraints.

Ant optimization and the ANTCOL algorithm

Ant optimization algorithms were introduced by Dorigo *et al*[10,11] and Dorigo[12] and belong to the class of population-based heuristics. They were inspired by the way that a colony of ants is able to identify the closest feeding source to their nest by cooperating with one another, by means of a pheromone trail. At the start of the search each ant sets off in a random direction until it finds food, when it returns to the nest, continuously laying a trail as it goes. As the trail evaporates over time, trails to further food sources will be weaker by the time the ant returns than those to closer sources. Subsequent ants tend to follow the trails left by their predecessors with a bias towards stronger trails. These are therefore reinforced further until eventually all the ants head towards the nearest food source. The idea behind the ant optimization algorithm is to use the concept of the trail to modify the probabilities of selecting the next option in a probabilistic greedy construction heuristic according to information on solution quality gained from previous solutions. As noted by Gambardella *et al*,[13] successful implementations were at first limited to travelling salesman[14] and other routing problems. These were followed by problems with natural representations as sequencing problems.[13,15] More recently, applications over a broader range of problems including graph colouring,[2,16] clique optimization,[17] frequency assignment[18] and university course timetabling[19] have been proposed. In the case of graph colouring Costa and Hertz[2] suggested the following implementation.

At each stage in a construction approach to the graph-colouring problem there are two decisions to be made: which vertex to colour next and what colour to use. A variety of properties have been suggested for making such decisions, for example, for selecting a vertex, the number of uncoloured neighbours, the number of colours already used on its neighbours. In ant optimization such measures are often referred to as the 'visibility' of an option, and a construction process in which an option is selected with a probability proportional to its visibility, will clearly result in a straightforward probabilistic greedy construction approach. In the first cycle of Costa and Hertz's ANTCOL algorithm a population of initial solutions is built in this way. The concept of the trail is then used to adapt the probabilities in subsequent cycles using information derived from previous cycles. Although the decisions in the construction process concern the desirability of colouring a particular vertex in a given colour, it is not sensible to store the trail information in this way. This is because the quality of a colouring does not depend on the absolute colour of each vertex, but on the sets of vertices that are placed in the same colour class. Therefore, Costa and Hertz define a trail matrix in terms of pairs of vertices. In the first cycle, the trail between each pair of non-adjacent vertices is set equal to 1. At the end of each cycle the trail matrix is updated according to $t_{ij} = \rho t_{ij} + \sum_{s \in S_{ij}} 1/q(s)$, where S_{ij} is the set of solutions in which i and j were coloured in the same colour, $q(s)$ is the number of colours required in solution s and ρ controls the rate of evaporation. During the construction process the trail associated with colouring vertex i in colour k depends on the set of vertices already coloured k, denoted V_k, and is given by $\tau_{ik} = \sum_{j \in V_k} t_{ij}/|V_k|$.

At each stage, the option of colouring vertex i in colour k is selected with probability

$$P_{ik} = \frac{\tau_{ik}^{\alpha}\eta_{ik}^{\beta}}{\sum_{(j,l)\in F} \tau_{ik}^{\alpha}\eta_{ik}^{\beta}} \quad \text{if}(i,k) \in F,\ P_{ik} = 0 \text{ otherwise}$$

where α and β are parameters that control the balance between the influence of the trail and visibility, and F is the set of feasible vertex-colour pairs. F will depend not only on the partial colouring at the current stage of the construction process but also on the underlying greedy construction algorithm. Eight variants of ANTCOL are suggested based on two of the most successful construction heuristics for graph colouring, recursive largest first (RLF) and saturation degree (DSATUR).

RLF constructs the colouring one colour class at a time, completing each class until no further additions are feasible before moving on to the next. Thus, the choice of colour is fixed and the only decision to be made is which vertex to select next. Three different rules are suggested giving rise to three different definitions of visibility η_{ik} for colouring vertex i in the current colour, k. These are defined as follows:

Let

W be the set of uncoloured vertices that can be included in the current colour class,

B be the set of uncoloured vertices that cannot be included in the current colour class,

$deg_X(i)$ = degree of vertex i in the subgraph induced by the subset of vertices, X.

Then the three options are (1) $\eta_{ik} = deg_B(i)$, (2) $\eta_{ik} = |W| - deg_W(i)$, (3) $\eta_{ik} = deg_{B \cup W}(i)$. As all three options depend on the vertices already coloured in the current colour, different rules are suggested for the first vertex in each colour. These are (1) randomly among those vertices for which $deg_W(i)$ is maximized, and (2) randomly in W. These rules combine to give six versions of the algorithm RLF(i,j) for $i = 1, 3$ and $j = 1, 2$ where i represents the choice of visibility function and j the rule for selecting the first vertex in each colour.

DSATUR works differently in that it does not complete a colour class before starting the next. It selects the next vertex to be coloured according to the saturation degree (ie the number of colours already used on its neighbours), and colours it in an appropriately chosen feasible colour. In the most common version of the algorithm the colours are considered in order and each vertex is coloured in the first feasible colour on the list. DSATUR is the basis of two variants of ANTCOL. In both η_{ik} is the saturation degree of vertex i for all k. In the first variant, the set of allowable colours is limited to the first available colour and a vertex is selected according to the above formula for P_{ik}. In the second, any feasible colour that has already been used is allowed. If no such colour exists then a new colour is considered. A vertex is selected first. This is done independently of the trail by setting $\tau_{ik} = 1\ \forall i, k$. Then a colour is selected from those feasible for that vertex with probability P_{ik} by setting $\eta_{ik} = 1$ and limiting the set F to pairs of the form (i^*, c) where i^* is the selected vertex and c is a colour that is feasible for i^*. These two variants will be denoted Dsat(1) and Dsat(2), respectively.

New evaluation functions and a diversification strategy

Although Costa and Hertz[2] obtain reasonable results with this heuristic, particularly with any of the RLF options using random selection of the first vertex in each colour, there is certainly room for improvement. In this section, we highlight some potential short comings of the basic version of ANTCOL and suggest possible improvements. As with all meta-heuristic approaches the performance of the algorithm will be influenced by the evaluation function used to measure solution quality, and in the case of ant optimization, used to determine the strength of the trail. In ANTCOL this is the number of colours required in an individual solution. This may not be a good measure as the range of possible values will tend to be small, especially as the use of visibility measures in the greedy construction process will tend to eliminate very bad solutions. This means that the differential in the trail factor added by good and not so good solutions will be small, especially as the number of colours increases. For example, for a graph that can be coloured in 40 colours the ratio of the trails added by a solution requiring just one extra colour to that added by solutions requiring two, three or four extra colours is just 1 : 0.976 : 0.953 : 0.932. For a graph requiring 20 colours this increases to 1 : 0.954 : 0.913 : 0.875 and for 10 colours to 1 : 0.917 : 0.846 : 0.785. This can be compensated to a certain extent using an appropriate value of α, but a different value might be required for different numbers of colours. In the case of examination scheduling the number of timeslots available and therefore the number of colours required are normally known. We can therefore measure solution quality as the number of *extra* colours used, and increase the trail by $1/(q(s)-r)$, where r is the number of colours available. This has the advantage that solutions that are x colours above the required number will have the same evaluation for all graphs, and means that the differential between a solution that requires just one extra colour and those requiring two, three or four extra colours is improved to 1 : 0.5 : 0.33 : 0.25.

Although the above modification increases the differential between good and not so good solutions it does not solve a second problem that arises with graph colouring—a large number of often diverse solutions requiring the same number of colours. Yet, intuitively we can see that some solutions are better than others. For example, given two solutions requiring $(r+1)$ colours, one in which the smallest colour class has just one vertex, and the other in which the colours are distributed evenly, there is a sense in which the first is closer to an optimal solution than the second. This problem has been recognized in the context of other meta-heuristic approaches to graph colouring and dealt with in a variety of ways. For example, in neighbourhood search techniques it is common to include a term of the form $-|V_k|^2$ in the cost function where V_k is the set of vertices coloured k. This will give preference to solutions with some large and some small colour classes over those with more even distributions. A similar point is made by Levine and Ducatelle[20] in their ant colony approach to the bin-packing problem where they include a term based on the sum of the percentage fill of the bins raised to a suitable power. In our case, the solution is simpler. As the construction process produces only feasible colourings and only adds a new colour class where necessary, we can limit the number of classes considered to the required number, r, and use the number of vertices uncoloured at the end of the construction process for solution s, denoted $u(s)$, as our evaluation function. However, when considered in the context of the way in which the ant algorithm operates, this may not be a good idea for the following reason. Recall that each solution increases the trail between all pairs of vertices coloured in the same colour. Thus any uncoloured vertices will not receive any increase in trail, and will be less likely to be chosen early in any colour class in future cycles. This is counter intuitive, as we need to make these 'difficult-to-colour' vertices more attractive so that they are coloured earlier in the process, before being blocked by their more appealing neighbours. We therefore introduce a further modification, whereby the trail factor between an uncoloured vertex and all other vertices is increased by a constant amount, c. $c=1/u(s)$ was used in all the experiments described in this paper. This can be regarded as an intelligent diversification strategy, in which the additional trail factor guides the search away from solutions in which the previously uncoloured vertices fail to be coloured by increasing their probability of being chosen.

Experiments

The objective of the first set of experiments was twofold. Firstly, to compare the performance of ANTCOL on typical timetabling graphs with that observed by Costa and Hertz for their set of random graphs, and secondly to determine the most promising combination(s) of construction heuristic, trail calculation and ANTCOL parameter values. In order to keep the computational effort required to a reasonable level these were based on five runs for each combination on some of the smaller examination timetabling problems from the public domain with up to 300 vertices. The best options were then tested further on a wider range of data sets. These experiments involved the original eight ANTCOL variants using trail strength determined by evaluation function $1/q(s)$, denoted ORIG, together with the three modifications outlined in the previous section, that is, using evaluation function, $1/q(s)-r$, denoted EXTRA, using evaluation function $1/u(s)$, denoted UNCOL and using evaluation function $1/u(s)$ with diversification, denoted DIVERS. These were run on the first seven data sets in Table 1. All experiments were run for 100 ants and 100 cycles using values of α and β ranging from 1 to 5, and a value of $\rho=0.5$. Each variant/trail combination was also run for all 5 values of β with no trail, corresponding to multiple starts with each of the randomized greedy construction heuristics, denoted NOTRAIL. Although for most of the data sets the number of slots available exceeds the number of colours in an optimal colouring, our experiments use $r=$ optimal number of colours. This is because we intend to test the graph-colouring ability of the algorithm under the most difficult conditions to highlight any differences in performance and to try to ensure that the performance is sufficiently strong to

Table 1 Examination Timetabling datasets from the public domain (ftp://ftp.mie.utoronto.ca/pub/carter/testprob)

Data set	*Abbreviation*	*Number of exams*	*Number of students*	*Number of timeslots*	*Optimum/best-known solution*
HEC-S-92	HEC	80	2823	18	17*
STA-F-83	STA	138	611	13	13
YOR-F-83	YOR	180	1125	21	19
UTE-S-92	UTE	184	2750	10	10*
EAR-F-83	EAR	189	941	24	22
TRE-S-92	TRE	261	4360	23	20*
LSE-F-91	LSE	381	2726	18	17*
KFU-S-93	KFU	461	5349	20	19*
RYE-F-92	RYE	487	11483	23	21*
CAR-F-92	CARF	543	18419	32	28
UTA-S-92	UTA	638	21266	35	30
CAR-S-91	CARS	682	16925	35	28

*Indicates that the value given is the optimum, as the best-known solution equals the size of the maximum clique. In other cases, the value given is the best-known solution.

form the basis of a complete examination scheduler. Each experiment was run five times using different random number seeds. Given the number of data sets this means that comparisons between different parameters, algorithm variants etc. will be based on at least 35 observations.

Costa and Hertz's performance measure is based on the mean of the best colouring found in each run. This is not an ideal measure for our analysis as the quality of options UNCOL and DIVERS is judged on the number of uncoloured vertices when only r colours are allowed, rather than the total number of colours required to colour everything. Nevertheless, we include results based on the mean in Figure 1a as one of our objectives is to compare our results with those of Costa and Hertz. Instead of the mean of the total number of colours required we use the mean excess over the optimal number of colours, that is, mean $(q(s)-r)$. This allows us to see immediately how close to the optimum the solutions are and also allows us to plot the mean number of uncoloured vertices for options UNCOL and DIVERS on the same scale. However we emphasize that the bars for the options UNCOL and DIVERS cannot be compared like-for-like with those of NOTRAIL, ORIG and EXTRA.

As our objective is to find feasible timetables in the minimum number of timeslots a more appropriate measure is the number of optimal completions. Two statistics relating to this measure are presented in Figure 1b and c. In Figure 1b the total number of optimal completions out of a total of 875 (seven data sets × five starts × 25 α/β combinations) is given for each trail calculation/construction heuristic combination. This gives an idea of the overall performance of each combination but does not highlight those options that may do well with the right α/β combination. Therefore, in Figure 1c, we illustrate the number of optimal completions (out of 35) for the best α/β combination in each case. (Note that we have chosen not to follow Costa and Hertz in presenting all the results for each α/β combination separately as this would result in an excessive number of tables. Instead, we highlight some general patterns and give a detailed

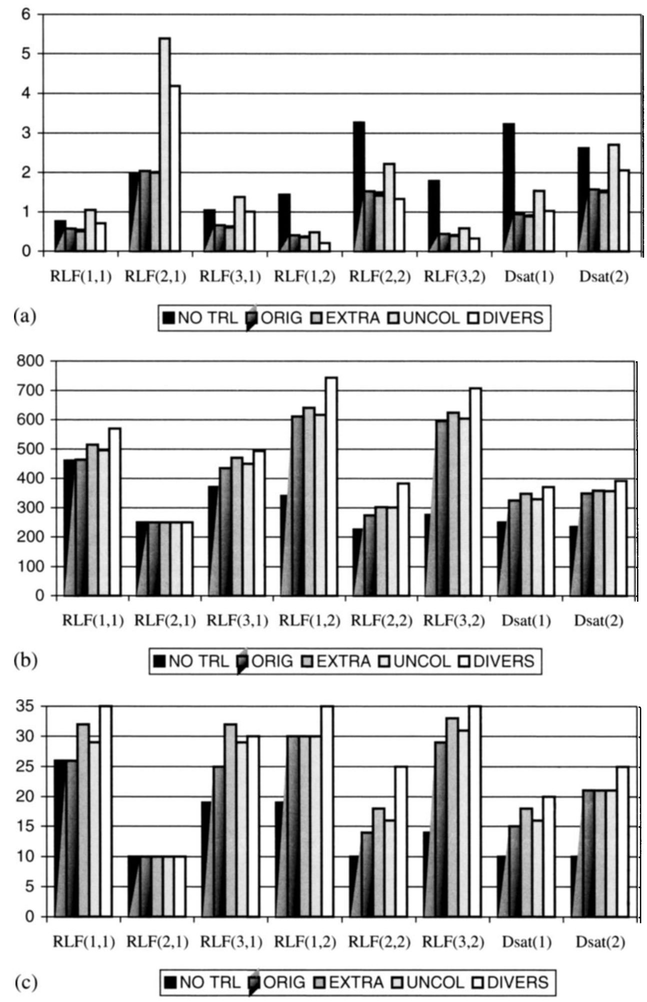

Figure 1 Performance of the different variants for all α/β combinations on data sets 1–7. (a) Mean number of excess colours for NO TRL, ORIG and EXTRA and mean number of uncoloured vertices for UNCOL and DIVERS. (b) Total number of optimal completions out of 875 (7 datasets × 25 α/β combinations × 5 random runs. (c) Number of optimal completions for the best α/β of 35 (7 datasets × 5 random runs).

breakdown for the more promising combinations later in this section).

A number of conclusions can be drawn from Figure 1. Firstly, there is a striking similarity between the shapes of Figure 1b and c. The relative performance of the different construction heuristics is also largely reflected in Figure 1a. The exception is the comparison of Dsat(1) and Dsat(2) where the difference is more pronounced in terms of mean solution quality than in the number of optimal completions, suggesting that when Dsat(2) fails to find the optimum it is likely to find worse solutions than Dsat(1).

It is also apparent that the construction heuristic has a significant impact on solution quality with the random starts variants of RLF (ie RLF(*,2)), performing better than their maximum degree counterparts (ie RLF(*,1)). This is in line with the findings of Costa and Hertz, as is the relatively poor performance of Dsat(2) in terms of the mean value. Where our results differ significantly from those of Costa and Hertz is in the results of RLF(2,*), the option based on deg_W, the number of uncoloured neighbours that are still feasible for the current colour. RLF(2,1) always solves data sets STA and UTE but fails to optimize any of the other data sets with any of the trail options. RLF(2,2) fairs a little better but is still a relatively poor performer. Costa and Hertz found little if any difference between the RLF options on their random graphs. Carter *et al*[8] suggest that random graphs tend to have a relatively small maximal clique whereas examination scheduling graphs frequently have a large clique of vertices of cardinality equal to or just short of the optimal number of colours. We have also found that many of these vertices also tend to have large degree. (For example, data set EAR has a maximum clique size of 21 vertices. The mean degree of these is 96.3 as compared to 44.9 in the graph as a whole.) In an optimal colouring one of these vertices must appear in all (or all but one or two) of the colour classes. Thus it is important that they are selected relatively early in each colour class before being barred by one of their neighbours. RLF(1,*) and RLF(3,*) are based on maximizing the degree of a vertex in subgraphs B and $B \cup W$, respectively, and will therefore be biased towards the selection of vertices of larger degree. Conversely, RLF(2,*) is based on minimizing the degree in subgraph W and will therefore be biased towards selecting vertices of lower degree.

Apart from showing the influence of the construction heuristic on solution quality Figure 1 also illustrates the impact of our modifications to the way in which the trail is calculated. With the exception of RLF(2,1) Figure 1b and c shows that the results from ORIG are improved by both EXTRA and UNCOL, with EXTRA performing slightly better than UNCOL. DIVERS, the option using UNCOL with a diversification strategy to overcome the potential problem of increasingly faint trails relating to uncoloured vertices that can occur with UNCOL, outperforms all three across the board. As stated earlier, Figure 1a cannot be used to compare ORIG and EXTRA with UNCOL and DIVERS, but it does show that EXTRA outperforms ORIG and DIVERS outperforms UNCOL across all construction approaches. Figure 1c shows that three variants—RLF(1,1)/DIVERS, RLF(1,2)/DIVERS and RLF(3,2)/DIVERS reach perfect scores of 35 optimal completions when run with the best α/β combination. Figures 1b and a suggests that of these RLF(1,2)/DIVERS is the strongest performer. Indeed, Figure 1a shows that the mean number of uncoloured vertices for this variant is smaller than the mean number of excess colours required by the best variant of ORIG or DIVERS. As the number of uncoloured vertices is an upper bound on the number of excess colours this confirms the superiority of RLF(1,2) DIVERS.

What these results do not show is the effect of α and β. This is addressed in Tables 2 and 3. Table 2 indicates which combinations gave the maximum number of successful completions over all 32 variants. An ‘x’ indicates that this combination gave the highest number of optimal completions for that variant, while a ‘–’ indicates that the combination gave just one less than the maximum. The number of optimal completions for that variant is given in the ‘no. max’ column to the right of the table. Table 3 gives the full breakdown for the three best options identified in the above. Both tables show that the best combination of parameters differs from variant to variant, but that a pattern can be observed, especially for those variants that are more successful. In Table 3, the results for RLF(2,1) can be discounted. These correspond to perfect solutions for the two easy data sets STA and UTE, and no other successes for all combinations of α and β. RLF(2,2) and Dsat(1) are also relatively poor performers and display a fairly random pattern. Within the remaining variants a number of patterns are apparent. For RLF(*,1) the best results are obtained with a trail factor exponent of $\alpha = 1$ and relatively high values of β, while for random starts RLF(*,2) the best results come from $\alpha = 2$, with a bias towards higher β values. Conversely, the less successful option of Dsat(2) shows a trend towards higher values of both α and β. It is also apparent that with the exception of the poorly performing variants the best solutions are rarely achieved for α greater than β.

These results suggest that a greedy construction heuristic based on RLF(1,2), a trail calculated according to DIVERS, and the right combination of α and β can produce good solutions for graphs derived from small to moderately-sized examination timetabling problems. In order to confirm these conclusions and to answer three further questions two additional sets of experiments based on this combination of construction heuristic and trail were carried out. The first was designed to confirm the consistent performance observed over five random starts and to measure how quickly an optimal solution could be found. In the light of a conclusion by Costa and Hertz, that the number of ants should be at least the number of vertices, runs using 100 ants

Table 2 The best α and β values for each variant. ('x' = combination with max. optimal completions, '–' = combination with max. −1)

	α	1					2					3					4					5					No.
	β	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	Max
Heuristic	Trail																										
RLF(1,1)	ORIG					x																					26
	EXTRA				x	–																					32
	UNCOL				x	–																					29
	DIVERS		x	x	x	x																					35
RLF(2,1)	ORIG	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10
	EXTRA	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10
	UNCOL	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10
	DIVERS	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10
RLF(3,1)	ORIG			–	x	x																					25
	EXTRA				x	x																					32
	UNCOL			–	x	x																					29
	DIVERS			x	–	–																					30
RLF(1,2)	ORIG							–		–	x																30
	EXTRA					–		–	x	x	x			–	–	x					–					–	30
	UNCOL									x	x																30
	DIVERS					x				x	x																35
RLF(2,2)	ORIG						–	–									–		x								14
	EXTRA						x										–										18
	UNCOL											–							–			x					16
	DIVERS						x																				25
RLF(3,2)	ORIG								x	x	x																29
	EXTRA								–	x																	33
	UNCOL									x																	31
	DIVERS							–	x	x	–																35
Dsat(1)	ORIG							x																			15
	EXTRA									–	x	–															18
	UNCOL						–	x		–	x		–	–		–				–	x						16
	DIVERS							–	x	–																	20
Dsat(2)	ORIG										–				–	x					–				–	–	21
	EXTRA									–	–				x	–				x	–				–	x	21
	UNCOL									–	–				–	–				–	–				–	x	21
	DIVERS															–					x					x	25

were compared with the results when the number of ants was set equal to the number of vertices. As data sets STA and UTE were solved optimally by all variants they were dropped for these experiments. The remaining five data sets in the original experiment were each run 100 times for $\alpha = 2$, $\beta = 4$ and $\beta = 5$. ($\alpha = 1$, $\beta = 5$ was not considered as the pattern observed in Table 2 suggested that this would not be as successful as the options with $\alpha = 2$). Each run was continued until an optimal solution was found or the maximum number of cycles reached 10 000/(number of ants). In each case the number of calls to the construction process was used to measure the computational effort required.

The second set of experiments were designed to measure the performance of the algorithm on larger data sets. Data sets 8–12 in Table 1 were run for five random starts using RLF(1,2)/DIVERS with $\alpha = 2$, and $\beta = 4$ and 5.

The results for the 100 run experiments on small to medium problems are shown in the first sections of Table 4. The three rows for each variant are the number of runs that failed to find the optimal solution, the mean number of calls to the construction heuristic and the mean time in seconds until an optimum solution was found or the maximum number of cycles was completed without reaching the optimum. Data sets HEC, EAR, TRE and LSE all solved optimally every time. YOR proved more difficult with a number of failures in each run. $\beta = 5$ produced more optimal completions than $\beta = 4$, while setting the number of ants equal to the number of vertices was better than a constant colony size of 100, with just six failures for the best set of options. In terms of computational effort $\beta = 5$ again outperformed $\beta = 4$, but for all except the difficult YOR data set using fewer ants produced quicker convergence. However, in view of the increased probability of finding optimal solutions for more difficult data sets the trade-off in terms of longer computation times on easier data sets is probably worthwhile. In the experiments on the larger graphs optimal solutions were produced in all instances except for one run with the UTA data set and $\beta = 5$.

Table 5 compares our results with those of other researchers who have tackled the same problem, that is, minimizing the number of timeslots required for a clash free

Table 3 Performance of three best options for each α/β combination

β	α 1	2	3	4	5
RLF(1,1)/DIVERS					
0	10 (1.46)	14 (1.00)	20 (0.54)	22 (0.37)	26 (0.26)
1	20 (0.63)	35 (0.0)	35 (0.0)	35 (0.0)	35 (0.0)
2	15 (1.09)	22 (0.49)	26 (0.37)	28 (0.23)	29 (0.17)
3	14 (1.63)	16 (0.91)	21 (0.60)	23 (0.43)	25 (0.46)
4	12 (2.17)	17 (1.17)	21 (0.66)	24 (0.51)	23 (0.51)
5	12 (2.42)	14 (1.31)	20 (0.89)	22 (0.63)	25 (0.54)
RLF(1,2)/DIVERS					
0	10 (2.51)	10 (1.71)	13 (1.31)	16 (0.97)	19 (0.66)
1	16 (1.43)	28 (0.48)	30 (0.20)	33 (0.58)	35 (0.0)
2	31 (0.11)	33 (0.06)	33 (0.06)	35 (0.0)	35 (0.0)
3	29 (0.17)	28 (0.23)	31 (0.11)	32 (0.09)	30 (0.14)
4	27 (0.29)	28 (1.20)	29 (0.17)	31 (0.11)	29 (0.17)
5	30 (0.23)	25 (1.31)	28 (0.20)	28 (0.23)	29 (0.20)
RLF(3,2)/DIVERS					
0	10 (2.74)	10 (2.06)	11 (1.57)	10 (1.49)	14 (1.09)
1	15 (1.86)	25 (1.09)	25 (0.74)	25 (0.54)	25 (0.97)
2	31 (0.11)	34 (0.03)	35 (0.0)	35 (0.0)	34 (0.03)
3	27 (0.29)	29 (0.17)	32 (0.09)	32 (0.09)	31 (0.14)
4	23 (0.43)	26 (0.29)	29 (0.20)	30 (0.14)	30 (0.14)
5	23 (0.43)	27 (0.29)	27 (0.29)	31 (0.11)	27 (0.26)

The first figure is the number of optimal completions. The number in parentheses is the mean performance in terms of uncoloured vertices.

Table 4 A comparison of the computational effort required to find optimal solutions for different parameters

		HEC	*EAR*	*TRT*	*YOR*	*LSE*
RLF(1,2)	Not opt	0	0	0	22	0
$\alpha=2, \beta=4$	Calls	197.2	324.9	624.3	2504.7	119.37
Nants = 100	Time	0.23	3.10	14.09	36.09	4.27
RLF(1,2)	Not opt	0	0	0	17	0
$\alpha=2, \beta=5$	Calls	171.85	281.84	461.41	231.4	100.8
Nants = 100	Time	0.20	2.36	8.72	27.27	3.55
RLF(1,2)	Not opt	0	0	0	8	0
$\alpha=2, \beta=2$	Calls	190.3	490.2	1081.7	1564.9	373.6
Nants = $\|V\|$	Time	0.23	4.55	21.54	14.57	13.27
RLF(1,2)	Not opt	0	0	0	6	0
$\alpha=2, \beta=5$	Calls	149.0	424.3	833.4	1557.3	298.1
Nants = $\|V\|$	Time	0.17	4.00	14.91	14.27	12.72
Ndsat(2)	Not opt	0	0	3	0	0
$\alpha=7, \beta=7$	Calls	260.4	748.5	464.2	1002.7	377.1
Nants = nv	Time	0.75	12.21	122.30	13.80	24.02

Not opt is the number of runs that failed to find the optimal solution out of 100.
Calls is the mean number of calls made to the construction heuristic.
Time is the mean time in seconds for the optimal solution to be found or for all cycles to be completed.

timetable, using the same data sets. Computing times are reported in CPU seconds on a 600 MHz Pentium III PC, and are the mean run times for 100 cycles. These results show that ANTCOL using RLF(1,2)/DIVERS with $\alpha=2$, $\beta=5$ produces the best-known solutions for all data sets, matching the performance of Caramia *et al*[21] and out-performing Carter *et al*[8] and Merlot *et al*[22] in terms of best solution found, and outperforming all three in terms of

Table 5 Comparison of results of Carter *et al*, Merlot *et al* and Caramia *et al*

Data set		*ANTCOL RLF(1,2)/DIVERS*	*Carter et al*	*Merlot et al*	*Caramia et al*
HEC	Best	17	17	18	17
	Range	17–17	17–18	—	17–18
STA	Best	13	13	13	13
	Range	13–13	13–13	—	13–13
YOR	Best	19	19	23	19
	Range	19–20	19–21	—	19–21
UTE	Best	10	10	11	10
	Range	10–10	10–10	—	10–10
EAR	Best	22	22	24	22
	Range	22–22	22–24	—	22–23
TRE	Best	20	20	21	20
	Range	20–20	20–23	—	20–23
LSE	Best	17	17	18	17
	Range	17–17	17–18	—	17–18
KFU	Best	19	19	21	19
	Range	19–19	19–20	—	19–20
RYE	Best	21	21	22	21
	Range	21–21	21–23	—	21–23
CARF	Best	28	28	31	28
	Range	28–28	28–32	—	28–32
UTA	Best	30	32	32	30
	Range	30–30	32–35	—	30–34
CARS	Best	28	28	30	29
	Range	28–28	28–35	—	28–32

consistency. Run times are mostly comparable with those of Caramia *et al*[21] on smaller and medium-sized problems but are somewhat longer on larger problems. However, Caramia *et al* terminated their algorithm if there was no improvement for a set number of iterations whereas our times are for 100 cycles. As all runs converged to the best-known solution well within a limit of 30 cycles, run times could be cut to less than a third.

Practical considerations

The last section dealt with the single objective of finding a timetable in the minimum number of timeslots, subject to no student being required to take two exams at the same time. Our final implementation based on the best combination of greedy construction strategy, trail definition and parameter values was consistently able to match the best-known solution across a range of data sets. As the number of slots available in most cases (see Table 1) is always higher than this, and in some cases considerably so, the performance is certainly good enough to support further development of a practical examination scheduler based on ANTCOL. A full treatment of such development is beyond the scope of this paper, but it is worth commenting on the feasibility of incorporating additional constraints/objectives in the light of our findings. Here we limit our discussion to the three that occur most often in practice: time-windows, seating capacity and second-order conflict.

Time-windows

As stated earlier time-window constraints can be incorporated directly into the graph-theoretic model by adding an appropriate set of dummy vertices. Thus, in theory the graph colouring procedure outlined in the previous sections should be able to cope with time-windows without further modification. However, as noted when discussing the poor performance of the RLF(2,*) construction options, it is important that vertices from large cliques are selected relatively early in each colour class. It was conjectured that the success of the RLF(1,*) and RLF(3,*) options was their bias towards vertices of higher degree that tended to correspond to those vertices in the larger cliques. By definition, the time-widow vertices will form a clique of cardinality equal to the number of colours available. As some slots have only a small number of examinations barred from taking place within them they will have relatively small degree. Thus it is possible that they may not be selected as early as they need to be in order to produce an optimal colouring. An obvious way of overcoming this is to restrict the selection of the first vertex in each colour to one of the timeslot vertices—either randomly or according to degree. Conversely, given the superior performance of those variants that selected the first vertex in each colour at random over those selecting the vertex of maximum degree, this may be counter productive. In order to obtain some feeling for the difficulties that might be encountered in the presence of time-windows and the relative merits of selecting time-window vertices first or otherwise, the results of using RLF(1,*)/DIVERS and selecting the first vertex from the set of timeslot vertices were compared with RLF(1,2)/DIVERS. As none of the data sets in Table 1 have time-window constraints the experiments were carried out on two recent data sets from University of Wales Swansea, UK, SWAN1 and SWAN2 with 313 and 332 vertices, respectively. RLF(1,2)/DIVERS failed to find schedules within the specified number of time periods, while both data sets solved consistently when the time-window vertices were selected first. Although we cannot draw definitive conclusions from experiments on just two data sets the results are encouraging for future developments.

Seating capacities

The simplest way of adding a capacity constraint to our version of ANTCOL is to bar a vertex from the current class once its size exceeds the remaining capacity. However, it is possible that such a restriction will be detrimental to algorithm performance. Although the number of seats available will limit the numbers of students who can sit an exam in any one time period there are very few real data sets in the public domain that include such restrictions. However, artificial seating capacities have been added to several data sets by Burke *et al.*[23] In order to acquire a feel for the likely success of the suggested approach RLF(1,2)/DIVERS were modified appropriately and run on the data sets given in Table 5. α and β were set to 2 and 5 as defined in the previous section. All problems solved successfully within the required number of timeslots.

Second-order conflict

Second-order conflict is the most challenging of the three features to incorporate into the algorithm and is likely to need a substantial amount of further development and experimentation. However, it is worth considering our results in the light of this objective. The results showed that approaches based on RLF(1,2) significantly outperformed those based on Dsat. This is not good news for the inclusion of second-order conflict as any construction based on an RLF approach will not lend itself to minimizing back-to-back exams for the following reason. The process continues to fill the current slot until no further exams can be added. This means that all remaining exams will clash with those in the previous slot. Even if the slots are not ordered according to their colouring order, for any two adjacent slots all exams in the one that was coloured second will have conflict with at least one exam in the slot that was coloured first. Thus a construction approach that is not restricted to filling one colour class before starting another is required. This suggests that an algorithm based on one of the Dsat options is a more natural starting point for such development and leads to the question as to whether it is possible to improve the performance of a Dsat-type approach up to or close to the level of RLF(1,2)/DIVERS. We consider two possible approaches. The first is an attempt at hybridizing a method based on a first or best available colour approach, as in Dsat, with a visibility measure based on deg_B. The second is motivated by our conjecture that a key feature of the more successful variants is a bias towards selecting vertices that belong to large cliques early in the search.

An obvious hybridization of Dsat(1) and RLF(2,*) is to replace the use of saturation degree, $deg_S(i)$, as the visibility for vertex i by $deg_{B(f(i))}(i)$ where $B(q)$ is the set of uncoloured vertices for which colour q is no longer feasible and $f(i)$ is the first available colour for vertex i. However, without further modification this will revert to RLF(1,2) for the following reason. Let Q be the number of colours used so far. For any unused colour, $q>Q$, $deg_{B(q)}(i)=0\ \forall i$. Hence, the visibility of any vertex with $f(i)>Q$ will be zero, and such vertices will not be selected while any uncoloured vertices with $f(i)\leqslant Q$ remain. Thus colour class 1 will be filled before any vertex is placed in colour class 2 and so on. We therefore adapt this approach to allow several colour classes to be 'open' at the same time as follows. If there is any i for which $f(i)>Q$ then we know that we will eventually need to start a new colour class. Therefore, it is unlikely to be detrimental to start it now. We restrict the choice of vertices to those satisfying $f(i)>Q$ and select one such vertex at random. (Note that the trail is not relevant in such a selection as the trail relating to an empty colour class will be 0.) We denote this version Nrlf(1). The equivalent modification of Dsat(2) is not feasible as we need to be able to define a colour class, q, to associate with each vertex in order to calculate the visibility based on some set $B(q)$. In Dsat(2) the vertex is selected before the choice of colour.

Although the decision to start a new colour class whenever $f(i)>Q$ was born of necessity, it is possible that such a decision is beneficial in biasing the construction towards the early selection of awkward vertices that are likely to belong to large cliques. Such a modification may also improve Dsat(1) and Dsat(2). We therefore introduce two further variants Ndsat(1) and Ndsat(2) in which a vertex with $f(i)>Q$ is selected at random if any such vertices exist. If not the process reverts to Dsat(1) or Dsat(2), respectively.

In order to measure the quality of these three modifications the initial experiments, using α and $\beta=1{,}5$ on the seven initial datasets with five random starts for each combination, were repeated for the three new construction approaches. The results are shown in Figure 2 with RLF(1,2) included as a basis for comparison. The results indicate that although none of the approaches are able to match RLF(1,2) on all three measures, Ndsat(2) does achieve 35 optimal runs with the right combination of α/β. The mean scores for this variant are surprisingly poor, but this turns out to be due to very poor performance for low values of α and β. The best solutions for this option all occur when both α and β are high. In order to investigate the performance of this option,

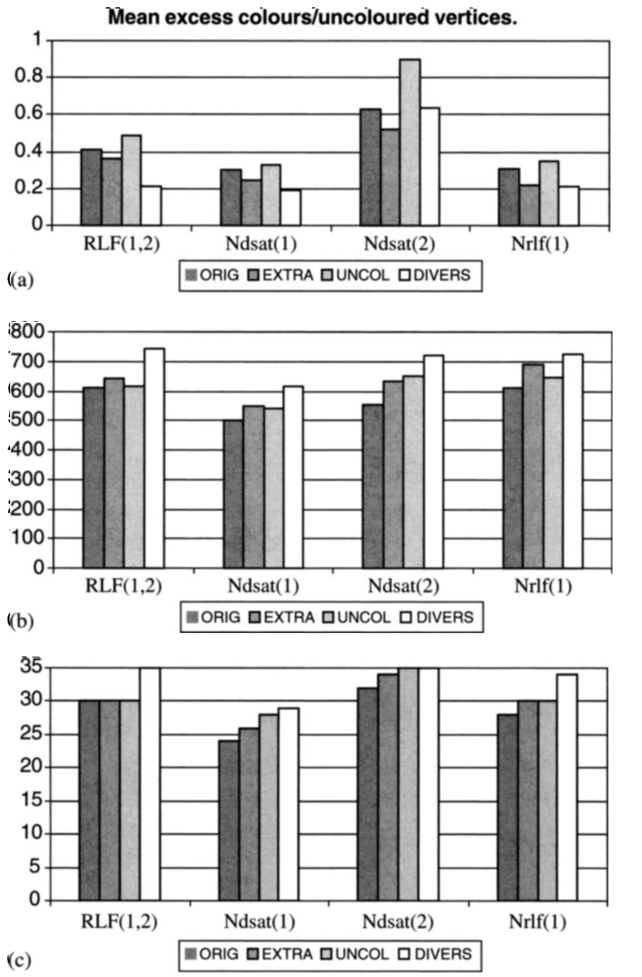

Figure 2 Performance of modified construction heuristics for all α/β combinations on data sets 1–7. (a) Mean number of excess colours for ORIG and EXTRA and mean number of uncoloured vertices for UNCOL and DIVERS. (b) Total number of optimal completions out of 875 (7 datasets $\times$ 25 α/β combinations $\times$ 5 random runs). (c) Number of optimal completions for the best α/β out of 35 (7 datasets $\times$ 5 random runs).

Ndsat(2)/DIVERS with $\alpha=\beta=5$ was run 100 times on the five non-trivial data sets. All data sets except TRE solved optimally for each run. The results for TRE were disappointing in that only 58 out of the 100 runs terminated optimally. As the results for the full range of α and β values show an improvement in solution quality as α and β increase, up to the current limit of five, the experiments were repeated using higher values of α and β. With $\alpha=\beta=6$, 82 runs converged optimally and with $\alpha=\beta=7$, only three runs failed to find the optimal solution. The full results for this option are shown in the last rows of Table 4. While this option is not as effective as RLF(1,2) in terms of the speed of convergence it is able to compete effectively in terms of solution quality, although the slow convergence of some of the TRE runs gives a mean solution time for this data set of over 2 min.

Future developments

In terms of examination scheduling, the results show that our modified version of ANTCOL, based on either RLF(1,2) or Ndsat(2) with the trail defined using the DIVERS option, is sufficiently powerful to encourage further development with the aim of producing a full timetabling system. For the reasons outlined in the previous section the Ndsat(2) option is the preferred starting point. However, the results also show that the construction heuristic has a significant influence on solution quality. This suggests that the construction heuristic in the full system will need to bias the search towards high-quality solutions in terms of both the graph colouring objective and the second-order conflict objective. In view of the direct conflict between these, a simple greedy approach may not be able to achieve this. In the light of this conclusion the following points are worth considering.

1. Some of the most successful examination scheduling algorithms to date have been based on local search approaches such as simulated annealing, tabu search, GRASP, etc[24–26] with those using neighbourhoods based on the graph theoretic concept of Kempe chains[23,27] being particularly powerful.
2. Some of the most successful ant optimization schemes include a local search element used to improve the solutions from the greedy construction phase, with information from the improved solution being fedback into the trail, for example, Bullnheimer *et al*,[28] Gambardella *et al*,[13] Stützle and Hoos.[29]

It therefore seems likely that while a successful implementation will require an appropriate balance between the primary and secondary objectives within the construction heuristic and the trail, a post-construction improvement phase may be required to reinforce the second-order conflict objective. Work to this end is already underway.

From a more generic viewpoint the modifications to the trail calculation introduced here are not restricted to the examination scheduling problem, and can be used in any situation where the objective is to find a feasible solution subject to a known and finite resource; for example in other forms of scheduling problems, cutting and packing problems, etc. These ideas can be generalized further. Rather than treat the target value, r, as a static quantity it can be treated as a dynamic moving target that improves as the results of the search improve, in much the same way as upper or lower bounds in a tree search. However, in our case as well as adjusting the evaluation function this will allow the use of the intelligent diversification strategy. In order to test the potential of this idea variant RLF(1,2)/DIVERS was modified so that the target, r, was initialized as being equal to the number of vertices, and reset at the end of each cycle as $r = Q^* - 1$, where Q^* is the minimum number of colours achieved to date. (In order to differentiate sufficiently between solutions that meet the target and those that just fail if $q(s) - r \leqslant Q$ then we set the increase in trail to be equal to 3). Data sets HEC, EAR, TRE, YOR and LSE were each run 100 times with this variant. With the exception of just one YOR run all terminated optimally, with the number of calls to the construction routine showing only a very small increase over that observed in the original fixed r version. This suggests that this approach might improve a classical ANTCOL algorithm for the general graph colouring problem and might also be successful for a range of other problems where the optimal solution is not known. This avenue of research is also being pursued.

Conclusions

This paper has added to the growing body of evidence to suggest that successful implementations of ant colony optimization need not be restricted to routing problems and their relatives that have been the main focus of research to date. Our improvements and modifications to the original ANTCOL algorithm for graph colouring, made with particular reference to the examination scheduling problem, resulted in an implementation that is competitive with the best-published approaches to the problem of minimizing the number of timeslots required for a clash-free timetable. We were also able to identify one variant that not only performs well with this objective but also displays characteristics that lend themselves to the inclusion of other commonly occurring constraints and objectives, including that of second-order conflict. Although the modifications to ANTCOL were motivated by the examination scheduling problem they can be applied directly to other problems where the objective is to find a feasible solution subject to a known resource constraint. It would be interesting to see if such modifications produce similar improvements for problems such as packing and other scheduling problems. We also suggested a further generalization to the case where the objective is to minimize the amount of resource required.

On a more abstract level our results show that, as with most other effective meta-heuristic implementations, success is dependent on getting the right balance between the intensity of the search in seeking out good solutions, and its diversity in allowing the exploration of new areas of the search space. In ant colony optimization the search for good solutions is incorporated in the underlying construction heuristic and in the learning process controlled by the definition of the trail. Our results suggest that the way in which these two factors are defined and the way in which they are weighted through the values of α and β all influence solution quality. The main source of diversity is the probabilistic nature of the selection process at each stage of the construction heuristic. However, the difference between RLF(*,1) (those variants that select a vertex of maximum degree to start each new colour class) and RLF(*,2) (those variants that select such vertices at random from those still available) shows that such diversity can also be introduced effectively into the construction heuristic itself. As well as these sources of background diversity our most successful variant included an intelligent diversification strategy implemented via the trail and designed to try to avoid some of the undesirable consequences of selections made in earlier cycles. The differences between the best and worst results over the different variants and parameter values suggest that, when developing a solution approach based on ant colony optimization, spending some time and effort on the design and evaluation of the various components, together with the way in which they interact with one another, is a worthwhile investment.

References

1 Colorni A, Dorigo M and Maniezzo V (1991). Distributed optimization by ant colonies. In: Varela FJ and Bourgine P (eds) *Proceedings of the First European Conference on Artificial Life*, Paris, MIT Press/Bradford Books, Cambridge, MA, pp 134–142.

2 Costa D and Hertz A (1997). Ants can colour graphs. *J Opl Res Soc* **48**: 295–305.

3 Costa D, Hertz A and Dubuis O (1995). Embedding a sequential procedure within an evolutionary algorithm for coloring problems in graphs. *J Heuristics* **1**: 105–128.

4 Vesel A and Žerovnik J (2000). How well can ants color graphs? *J Comput Inf Technol—CIT* **8**: 131–136.

5 Petford AD and Welsh DJA (1989). A randomised 3-colouring algorithm. *Discrete Math* **74**: 253–261.

6 Carter MW (1986). A survey of practical applications of examination timetabling algorithms. *Opns Res* **34**: 193–202.

7 Ross P, Hart E and Corne D (1998). Some observations about GA based exam timetabling. In: Burke E and Carter M (eds). *Practice and Theory of Automated Timetabling II—Lecture Notes in Computer Science*, Vol. 1408. Springer-Verlag, Berlin, pp 115–129.

8 Carter MW, Laporte G and Lee SY (1996). Examination timetabling: algorithmic strategies and applications. *J Opl Res Soc* **47**: 373–383.

9 Balakrishnan N (1991). Examination scheduling—a computerised application. *OMEGA* **19**: 37–41.

10 Dorigo M, Maniezzo V and Colorni A (1991). *Positive feedback as a search strategy*. Technical Report 91-016, Dipartimento di Elettronica e Informazione, Policecnico di Milano, Italy.

11 Dorigo M, Maniezzo V and Colorni A (1996). The ant system: optimization by a colony of co-operating agents. *IEEE Trans Systems Man Cybernetics—Part B* **26**: 29–41.

12 Dorigo M (1992). *Ottimazzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimisation, learning and natural algorithms)*. Doctoral dissertation, Dipartimento di Elettronica e Informazione, Policecnico di Milano, Italy.

13 Gambardella LM, Taillard ED and Dorigo M (1999). Ant colonies for the quadratic assignment problem. *J Opl Res Soc* **50**: 167–176.
14 Gambardella LM and Dorigo M (1996). Solving symmetric and asymmetric TSP's by ant colonies In: *Proceedings of the IEEE Conference on Evolutionary Computation (ICEC'96)*. IEEE Press, New York, pp 622–627.
15 Colorni A, Dorigo M, Maniezzo V and Trubian M (1994). Ant system for Job-shop Scheduling. *JORBEL—Belgian J Opns Res, Statist Comput Sci* **34**: 39–53.
16 Comellas F and Ozón J (1998). An ant algorithm for the graph colouring problem. Presented at *Ants 98 From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimization*—Brussels, Belgium.
17 Fenet S and Solnon C (2003). Searching for maximum cliques with ant colony optimisation. In: Raidl G *et al* (eds). *Applications of Evolutionary Computing—Lecture Notes in Computer Science*, Vol. 2611. Springer-Verlag, Berlin, pp 236–245.
18 Montemanni R, Smith DH and Allen SM (2002). An ants algorithm for the minimum-span frequency-assignment problem with multiple interference. *IEEE Trans Vehicular Technol* **51**: 949–953.
19 Socha K, Sampels M and Manfrin M (2003). Ant algorithms for the university course timetabling problem with regard to the state of the art. In: Raidl G *et al* (eds). *Applications of Evolutionary Computing—Lecture Notes in Computer Science*, Vol. 2611. Springer-Verlag, Berlin, pp 334–345.
20 Levine J and Ducatelle F (2004). Ant colony optimisation and local search for bin packing and cutting stock problems. *J Opl Res Soc* **55**: 705–716.
21 Caramia M, Dell'Olmo P and Italiano GF (2001). New algorithms for examination timetabling. In: Näher S and Wagner D (eds). *Algorithm Engineering 4—Lecture Notes in Computer Science*, Vol. 1982. Springer-Verlag, Berlin, pp 230–241.
22 Merlot L, Boland N, Hughes B and Stuckey PJ (2003). A hybrid algorithm for the examination timetabling problem. In: Burke E and De Causemacker P (eds). *Practice and Theory of Automated Timetabling IV—Lecture Notes in Computer Science*, Vol. 2740. Springer-Verlag, Berlin, pp 207–231.
23 Burke EK, Newall J and Weare RF (1996). A memetic algorithm for university exam timetabling. In: Burke E and Ross P (eds). *Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science*, Vol. 1153. Springer-Verlag, Berlin, pp 241–250.
24 Thompson JM and Dowsland KA (1998). A robust simulated annealing based examination timetabling system. *Comput Ops Res* **25**: 637–648.
25 Di Gaspero L and Schaerf A (2000). Tabu search techniques for examination timetabling. In: Burke E and Erben W (eds). *Practice and Theory of Automated Timetabling III. Lecture Notes in Computer Science*, Vol. 2079. Springer-Verlag, Berlin pp 104–117.
26 Casey S and Thompson JM (2003). GRASPing the examination scheduling problem. In: Burke E and De Causemacker P (eds). *Practice and Theory of Automated Timetabling IV—Lecture Notes in Computer Science*, Vol. 2740. Springer-Verlag, Berlin, pp 232–244.
27 Thompson JM and Dowsland KA (1996). Variants of simulated annealing for the examination timetabling problem. *Ann Opns Res* **63**: 105–128.
28 Bullnheimer B, Hartl RF and Strauss C (1999). Applying the ant system to the vehicle routing problem. In: Voss S, Martello S, Osman IH and Roucairol C (eds). *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Dordrecht, pp 285–296.
29 Stützle T and Hoos H (1999). MAX–MIN ant system and local search for combinatorial optimisation problems. In: Voss S, Martello S, Osman IH and Roucairol C (eds). *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer, Dordrecht, pp 313–329.

Received October 2003;
accepted May 2004 after one revision