Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 35 (2008) 2728–2749

computers & operations research

www.elsevier.com/locate/cor

Solution bias in ant colony optimisation: Lessons for selecting pheromone models

James Montgomery[a,*,1], Marcus Randall[b], Tim Hendtlass[a]

[a] Faculty of Information and Communication Technologies, Swinburne University of Technology, Vic 3122, Australia
[b] School of Information Technology, Bond University, Qld 4229, Australia

Available online 25 January 2007

Abstract

Ant colony optimisation is a constructive metaheuristic in which solutions are built probabilistically influenced by the parameters of a pheromone model—an analogue of the trail pheromones used by real ants when foraging for food. Recent studies have uncovered the presence of biases in the solution construction process, the existence and nature of which depend on the characteristics of the problem being solved. The presence of these solution construction biases induces biases in the pheromone model used, so selecting an appropriate model is highly important. The first part of this paper presents new findings bridging biases due to construction with biases in pheromone models. Novel approaches to the prediction of this bias are developed and used with the knapsack and generalised assignment problems. The second part of the paper deals with the selection of appropriate pheromone models when detailed knowledge of their biases is not available. Pheromone models may be derived either from characteristics of the way solutions are represented by the algorithm or characteristics of the solutions represented, which are often quite different. Recently it has been suggested that the latter is more appropriate. The relative performance of a number of alternative pheromone models for six well-known combinatorial optimisation problems is examined to test this hypothesis. Results suggest that, in general, modelling characteristics of solutions (rather than their representations) does lead to the best performance in ACO algorithms. Consequently, this principle may be used to guide the selection of appropriate pheromone models in problems to which ACO has not yet been applied.

Keywords: Ant colony optimisation; Constructive metaheuristic; Pheromone model; Solution bias

1. Introduction

Metaheuristic search algorithms are intended to explore solution space in a purposeful manner so that good solutions are found. In general terms, *solution bias* is the tendency for a search algorithm to produce (i.e., find) some solutions more frequently than others and can thus be seen as a critical feature of such algorithms.

ACO is a constructive metaheuristic that belongs to the class of model-based search (MBS) algorithms [1]. In such algorithms, new solutions are generated using a parameterised probabilistic model, the parameters of which are updated using previously generated solutions so as to direct the search towards promising areas of the solution space. The model

* Corresponding author. Fax: +61 3 9819 0823.
E-mail addresses: jmontgomery@swin.edu.au (J. Montgomery), mrandall@bond.edu.au (M. Randall), thendtlass@swin.edu.au (T. Hendtlass).
1 This research was conducted while the author was at the School of Information Technology, Bond University, whose support he gratefully acknowledges.

0305-0548/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2006.12.014

used in ACO is known as *pheromone*, an artificial analogue of the chemical used by real ants to mark trails from the nest to food sources. Just as the intensity of pheromone on alternative path segments influences the way real ants travel from their nest to food, the parameters of a *pheromone model* influence the solution components (artificial) ants select to build solutions. As with any metaheuristic, ACO must be adapted to suit each particular problem to which it is applied, with the pheromone model being a key component of this process [2,3]. Often a number of alternative models may be used with a particular problem, so selecting the most effective is an important goal.

A number of recent studies have shown that the solution construction process and, consequently, the pheromone model used may exhibit an unwanted and potentially unfavourable bias [4–9]. It would be preferable to select a pheromone model for a problem in which bias is either minimised or most favourable (i.e., good solutions are more likely to result). Accordingly, understanding the different biases that alternative pheromone models will exhibit is essential to the future application of ACO.

The first part of this paper reviews the nature of bias in the constructive process and presents new findings that show how these biases lead to bias in the pheromone models used in ACO. Sections 2 and 3 serve as an introduction to the solution construction process and how pheromone models may be derived from a problem model, respectively. Section 4 summarises new findings concerning the interaction of biases due to solution construction and different pheromone models, while Section 5 introduces some novel techniques for the analysis and prediction of bias in pheromone models applied to particular problems. Based on these findings, Section 6 discusses issues that must be considered when selecting a pheromone model for a problem.

It has been suggested that, in the absence of specific knowledge of bias in a given pheromone model for a problem, a pheromone should be chosen that models solution identity rather than the structure of solutions in the constructive algorithm used [10]. The second part of the paper (Section 7) presents an empirical study that tests this hypothesis on six well-known combinatorial problems. As a number of different combinatorial optimisation (CO) problems are discussed this paper will be of most benefit to those with some familiarity with those problems. However, brief descriptions of each problem are also provided. Section 8 summarises the findings with regards to the prediction of bias and empirical performance of different pheromone models.

2. Solution construction and constructed solution biases

Constructive heuristics take an empty solution $s = \emptyset$ and successively add *solution components* to build a complete, typically feasible, solution to a problem. Denote the set of solution components by $\mathfrak{C}$, and a single solution component by $\mathfrak{c}_i \in \mathfrak{C}$. As constructive heuristics typically add solution components one at a time, a solution may be represented as a sequence of solution components $\mathfrak{s} = \langle \mathfrak{c}_i, \dots, \mathfrak{c}_k \rangle$. Partial sequences are denoted by $\mathfrak{s}^\mathrm{p}$, while a set of sequences is denoted by $\mathfrak{S}$. At each step of a constructive heuristic, the set of solution components that may be added to the partial solution $\mathfrak{s}^\mathrm{p}$ is given by $\mathfrak{N}(\mathfrak{s}^\mathrm{p}) \subseteq \mathfrak{C}$. The constructive algorithm used to build solutions defines a mapping from sequences of solution components to solutions. The set of sequences that represent a solution s is denoted by $\mathfrak{S}(s)$.

The nature of the solution components used depends on the problem specification and, to a lesser extent, the preferences of the constructive heuristic's designer. For instance, in the well-known travelling salesman problem (TSP) [11], where solutions may be represented as permutations of the cities that must be visited, $\mathfrak{C}$ may be defined to represent the set of cities. A sequence that is a permutation of the elements of $\mathfrak{C}$ will then correspond to a valid solution.

Note that the usage of the term solution component in this paper differs from that in some of the ACO literature (e.g., [4,12]), in which it is defined in terms of the model used by ACO. For instance, Blum [13] describes the set of cities in the TSP as *natural solution components*, because sequences are most naturally built from these kinds of components, although the addition of a single component $\mathfrak{c}_j$ to a partial sequence ending with $\mathfrak{c}_i$ actually corresponds to the addition of the edge "solution component" $(\mathfrak{c}_i, \mathfrak{c}_j)$. As it is useful to discuss the biases that result from the way in which solutions are built separately from biases in the model used in an ACO algorithm, a different term is introduced in Section 3 to refer to components *of the model*.

Recently, a number of authors have recognised that the *sequence space* that constructive algorithms explore forms a tree of constructive decisions [8,14–16]. Montgomery et al. [8] refer to this tree as a *construction tree*, denoted by $\mathscr{T}$. The root of a construction tree corresponds to the empty sequence $\langle\rangle$ and consequently to the solution $\emptyset$, while leaves correspond to completed or infeasible partial sequences and so to completed or infeasible partial solutions.

Considering the application of an *undirected* constructive algorithm that makes each constructive decision probabilistically using a uniform random distribution over the components available at each step (referred to hereafter as $\mathrm{ACO}_{\mathrm{undir}}$), Montgomery et al. [8] define two types of bias that may exist in a constructive algorithm for a particular problem. The first occurs when the mapping from sequences of solution components to solutions is non-uniform, i.e., some solutions are represented more than others in the space of sequences ants search. This bias is consequently called a *representation bias*.

Definition 1. A constructive algorithm applied to a given CO problem is said to have a *representation bias* if there exist two solutions s_1 and s_2, $s_1 \neq s_2$ such that $|\mathfrak{S}(s_1)| \neq |\mathfrak{S}(s_2)|$. If the probability of producing each sequence representing s_1 or s_2 is the same, $|\mathfrak{S}(s_1)| > |\mathfrak{S}(s_2)| \rightarrow P(s_1) > P(s_2)$, where $P(s)$ is the probability of $\mathrm{ACO}_{\mathrm{undir}}$ producing solution s.

The second occurs when problem constraints make certain sequences infeasible, so certain component choices restrict available options later in solution construction more than other choices. Sequences in which choices are limited have an increased probability of being produced as there are fewer opportunities during construction for the algorithm to produce a different solution. As this is a feature of constructing a solution sequentially, this bias is called a *construction bias*.

Definition 2. A construction tree $\mathscr{T}$ has a *construction bias* if there exist two nodes in $\mathscr{T}$ such that their heights in the tree are equal yet their degrees are not equal. A construction bias favours sequences on paths in $\mathscr{T}$ with fewer alternative branches than other paths in $\mathscr{T}$.

Collectively, the two biases are referred to as *constructed solution biases*. Different problems (and different approaches to building solutions) exhibit different combinations of the two biases. For instance, solutions to the TSP and quadratic assignment problem (QAP)[2] can be represented as permutations of the cities and locations involved, respectively. As each solution is represented by the same number of sequences and each sequence corresponds to a feasible solution these problems consequently have no constructed solution bias.

In contrast, the job-shop scheduling problem (JSP),[3] where solutions are represented by permutations of the operations to be scheduled, typically has both a representation bias (i.e., there is a non-uniform many-to-one mapping from sequences to solutions) and a construction bias (as problem constraints exclude certain sequences from being produced). The open shop scheduling problem (OSP) (see, e.g., [5]) is similar to the JSP, except that operations within each job may be processed in any order. Accordingly, it has a representation bias as in the JSP, but has no construction bias as all permutations of operations correspond to feasible solutions.

Constructed solutions to the multiple knapsack problem (MKP)[4] are typically sequences of the items chosen. They have both a representation bias (as to each solution of k items there correspond $k!$ sequences) and a construction bias (as some combinations are infeasible). Moreover, solutions can be of variable size, which is a particular kind of construction bias in which *shorter* solutions have an increased probability of being produced. Constrained assignment problems such as the generalised assignment problem (GAP)[5] also have variable length solutions, except that short solutions correspond to infeasible partial solutions. The distribution and number of these infeasible solutions can be altered in assignment problems by changing the order in which items are assigned, which is discussed in detail by Montgomery [7].

The presence of constructed solution biases in some problems raises the question of whether pheromone models can be chosen for those problems which are not also biased. However, before this question can be answered, it is necessary to define pheromone models more formally.

[2] The QAP is typically posed in terms of assigning n facilities to n different locations such that the costs of commodity flows between facilities is minimised.

[3] The JSP consists of determining the processing order of a number of operations (which are partitioned into different jobs) on a number of machines, where the processing order of operations within each job is fixed.

[4] The MKP consists of selecting a maximally valued subset of items (each with an associated value and set of resource requirements) from some set such that resource utilisation constraints are respected.

[5] The GAP consists of assigning each of n tasks to exactly one of m agents, where an agent may be assigned zero or more tasks, subject to agents' respective capacities.

3. Describing pheromone models

A pheromone model consists of a set of *solution characteristics*, where each solution characteristic is some identifiable feature of a solution that may or may not be present [10]. To each solution sequence or solution there corresponds a set of solution characteristics that is a subset of that defined by the model. A pheromone value, typically a non-zero positive real number, is associated with each solution characteristic, indicating the learned utility of having that solution characteristic in a solution. At each step of solution construction, an ant's choice of solution component is in effect a choice over one or more solution characteristics that will be present after that component is chosen. An ant's decision is guided by solution characteristics' pheromone values, which are adjusted at the end of each construction iteration by increasing the pheromone value for solution characteristics in the best solution(s) in proportion to their quality.

As discussed in Section 2, the definition of solution characteristic given here is synonymous with that of solution component in some of the ACO literature. The term solution characteristic has been introduced as it is often convenient to define solution characteristics in terms of the low-level components from which solutions are built, possibly in combination with other problem entities [7,10]. Additionally, bias due to solution construction and bias present in a pheromone model, while related, can only be discussed separately if a distinction is made between components of constructed solutions and components of the model used to influence construction.

Denote a pheromone model (i.e., set of solution characteristics) by C. Considering the pheromone model typically used with the TSP, which consists of a set of edges between cities, $C \subset \mathfrak{C} \times \mathfrak{C}$, where $\mathfrak{C}$ is the set of cities. In the MKP, where the typical pheromone model consists of the set of items, $C = \mathfrak{C}$, where $\mathfrak{C}$ is the set of items.

When the inclusion of a single solution component corresponds to the addition of a single solution characteristic from the model, the pheromone is considered to be *first order* [2,10,13]. For example, in the TSP, adding city $\mathfrak{c}_j$ to the partial solution sequence $\langle \mathfrak{c}_a, \mathfrak{c}_b, \ldots, \mathfrak{c}_i \rangle$ adds the single solution characteristic $(\mathfrak{c}_i, \mathfrak{c}_j)$. Higher order pheromone models occur when the addition of a solution component relates to a number of solution characteristics. For example, in the MKP a pheromone model $C \subset \mathfrak{C} \times \mathfrak{C}$ could be used to represent pairs of items from $\mathfrak{C}$ being copresent in a solution. The addition of a single solution component to a partial solution then corresponds to adding multiple solution characteristics, one for each other solution component already present in the solution. A number of higher order pheromone models are described and tested in Section 7. Where the same set of component combinations may be used to model qualitatively different characteristics—as in the first order model for the TSP and higher order model for the MKP, in which $C \subset \mathfrak{C} \times \mathfrak{C}$—it is necessary to provide a description of the correct interpretation of the characteristics modelled, such as $\mathfrak{C} \times \mathfrak{C}$ (copresent) for the higher order model for the MKP.

The pheromone models discussed in this paper use a number of different entities, summarised in Table 1, depending on the problem with which they are used. Assignment problems such as the QAP and GAP can be discussed in terms of two generic types of entities, *items* and *resources*, where each item must be assigned exactly one resource and each resource may be assigned zero or more items, depending on the problem. In the QAP, items typically represent facilities while resources represent locations (although the two are semantically equivalent in that problem), while in the GAP items represent tasks and resources represent agents. As each item must be assigned a resource, it is typical that solutions are represented as sequences of the resources, with each sequence position associated with an item.

Although the nature of solution characteristics modelled will be partially determined by the nature of the problem being solved, there are often a number of alternative models available for a given problem. For instance, three pheromone models have been used with the JSP, with solutions represented by permutations of the operations. Colorni et al. [17] model the placement of one operation immediately after another in the solution sequence, i.e., $C \subset \mathfrak{C} \times \mathfrak{C}$, in the same

Table 1
Pheromone model entities

Symbol	Entity represented
$\mathfrak{C}$	Solution components, e.g., items, cities
$\mathfrak{C}_{\mathrm{it}}$	Items in an assignment problem, e.g., tasks in the GAP
$\mathfrak{C}_{\mathrm{res}}$	Resources in an assignment problem, e.g., agents in the GAP
P	Absolute position of a solution component in a solution sequence

manner as pheromone models for the TSP. Others (see, e.g., [2]) have modelled the absolute position of operations within the sequence, $C \subset \mathfrak{C} \times P$. Blum and Sampels [2,5,18] use a second order pheromone which models the *relative* order of pairs of operations that require the same machine. Hence, the pheromone model is $C = \{(\mathfrak{c}_i, \mathfrak{c}_j) \in \mathfrak{C} \times \mathfrak{C} \mid \mathfrak{c}_i \neq \mathfrak{c}_j, M(\mathfrak{c}_i) = M(\mathfrak{c}_j)\}$, where $M(\mathfrak{c}_i)$ is the machine required by operation $\mathfrak{c}_i$, and each decision about the addition of a single operation $\mathfrak{c}_i \in \mathfrak{C}$ is influenced by multiple values from the set defined by $\{(\mathfrak{c}_i, \mathfrak{c}_j) \in C, \mid \mathfrak{c}_j \notin \mathfrak{s}\}$.

3.1. Representation-oriented and identity-oriented pheromones

Two approaches may be taken to the derivation of a pheromone model from the problem model used by ants to build solutions: representation-oriented and identity-oriented [10]. The first produces pheromones that reflect some aspect of *how* solutions are represented (i.e., the arrangement of solution components in a solution sequence), while the second results in pheromones that describe *which* solutions are represented. These are specified more formally in Definitions 3 and 4.

Definition 3. A pheromone model C is said to be *representation-oriented* if, for each sequence $\mathfrak{s}$, there is a unique set of solution characteristics drawn from C that describes only $\mathfrak{s}$.

Definition 4. A pheromone model C is said to be *identity-oriented* if, for each solution s, there is a unique set of solution characteristics drawn from C that describes s.

The two kinds of pheromone are not mutually exclusive. For instance, $\mathfrak{C} \times \mathfrak{C}$ pheromone for the TSP represents both how solutions are represented (i.e., as permutations) and the identity of solutions as sets of edges.

Given that pheromone models are related to sequences, solutions or both, and that it is on these that constructed solution biases act, there is necessarily an interaction between a model and these underlying biases.

4. Constructed solution biases and pheromone models

Different pheromone models interact with the same CO problem in different ways as the different solution characteristics they model correspond to different patterns of arcs in the construction tree. Montgomery [7] notes that the interaction of pheromone and constructed solution biases cannot be analysed precisely for any non-trivial problem instance, as it becomes impossible to perform a complete exploration of the construction tree. While the constructed solution biases are still present, as the underlying mechanisms do not change with problem size, any constructive search algorithm can at best produce a sample of the many feasible solutions to such instances, so a different approach is required to understand the nature of the interaction.

As problem instance size grows the number of sequences representing solutions $|\mathfrak{S}|$ grows at least as fast as the growth in solutions, which is typically exponential. In problems with a representation bias $|\mathfrak{S}|$ will grow even faster. However, the number of solution characteristics modelled by a pheromone model $|C|$ grows at a slower rate. For instance, for the MKP, $|\mathfrak{S}| = O(n!)$, while $|C| = O(n)$, where n is the number of items. In the GAP $|\mathfrak{S}| = O(m^n)$, while $|C| = O(m \cdot n)$, where n is the number of tasks and m is the number of agents. Consequently, the average number of sequences in which each solution characteristic appears (denoted by $|\mathfrak{S}_\mathrm{c}|$) grows with instance size and any differences between $|\mathfrak{S}_\mathrm{c}|$ for different solution characteristics will influence which are most likely to be reinforced by solutions produced by the algorithm.

To identify which pheromone models will or will not be biased, Blum and Dorigo [4] introduce the concept of a *competition balanced system* (CBS). In terms of ACO, this is defined as a pheromone model consisting of solution characteristics that appear in the same number of sequences produced by the algorithm. Blum and Dorigo [4, p. 163] provide the following definition of a CBS (explanatory notes appear in square brackets):

> Given a model $\mathscr{P}$ of a CO problem, we call an ACO algorithm and a problem instance P of $\mathscr{P}$ a **CBS**, if the following holds: given a feasible partial solution $\mathfrak{s}^\mathrm{p}$ and the set of solution components [equivalent to a solution characteristic as defined in Section 3] $\mathfrak{N}(\mathfrak{s}^\mathrm{p})$ that can be added to extend the partial solution $\mathfrak{s}^\mathrm{p}$, each solution component [i.e., characteristic] $\mathfrak{c} \in \mathfrak{N}(\mathfrak{s}^\mathrm{p})$ is a component of the same number of feasible solutions (in terms of sequences built by the algorithm) as any other solution component $\mathfrak{c}' \in \mathfrak{N}(\mathfrak{s}^\mathrm{p})$, $\mathfrak{c} \neq \mathfrak{c}'$.

Note that the term *solution component* in this definition is equivalent to the definition of a *solution characteristic* defined in Section 3. If a pheromone model applied to a particular problem instance is not a CBS, bias may be observed. The definition of a CBS does not explicitly incorporate the influence of the underlying constructed solution biases, although the presence of such biases is the underlying determinant of whether or not a pheromone model is a CBS. Furthermore, it is possible for a pheromone model to be a CBS, yet not be free from bias.

Theorem 5. *A CBS* (*as defined by Blum and Dorigo* [4]) *is not necessarily free from solution bias*.

Proof. When constructing solutions to the OSP as permutations of operations, both $\mathfrak{C} \times \mathfrak{C}$ and $\mathfrak{C} \times P$ pheromone models are CBSs, as each solution characteristic appears in the same number of sequences (since all permutations of operations are feasible). However, the OSP also has a representation bias and hence some solutions will be represented by larger sets of solution characteristics, corresponding to each sequence representing those solutions. The resulting pheromone–problem combination is therefore not free from solution bias. □

The following lemmas demonstrate that, for all practical purposes in other circumstances, constructed solution biases will prevent any pheromone model from being a CBS.

Lemma 6. *A representation-oriented pheromone model applied to a CO Problem with a construction bias cannot be a CBS*.

Proof. A representation-oriented pheromone model (as defined in Section 3.1) contains solution characteristics that describe each sequence uniquely. Consequently, in order that such a pheromone model be a CBS, each sequence must represent a feasible solution. This criterion is not met by a problem with a construction bias. □

Lemma 7. *An identity-oriented pheromone model applied to a CO Problem with a representation bias cannot be a CBS*.

Proof. An identity-oriented pheromone model (as defined in Section 3.1) contains one set of solution characteristics for each distinct solution. In order that such a pheromone model be a CBS, each solution must be represented by the same number of sequences. This criterion is not met by a problem with a representation bias. □

Lemma 8. *An identity-oriented pheromone model applied to a CO Problem with a construction bias cannot be a CBS*.

Proof. Given Lemmas 6 and 7, Lemma 8 need only be proved for those cases where the pheromone is not also representation-oriented (see Lemma 6) and where each solution is represented by the same number of sequences (see Lemma 7). The latter implies that each solution is described by the same number of solution characteristics (otherwise the problem must have a representation bias due to the different number of permutations of solution characteristics that would result if solutions were described by a variable number of characteristics). In order that such a pheromone be a CBS, every solution characteristic must appear in the same number of combinations (of solution characteristics) as every other. A construction bias implies that this criterion cannot be met, as some solution characteristics must appear less often than others for the degree of nodes within each level of the construction tree to be non-uniform. □

Based on the above, it is possible to state the following theorem.

Theorem 9. *If a constructive algorithm applied to a CO Problem has a construction bias then there does not exist a representation- or identity-oriented pheromone representation that is a CBS*.

Proof. The statement is a direct consequent of Lemmas 6 and 8. □

Given these findings, it is plausible that if a problem has no constructed solution bias then for all practical cases any pheromone model for it will be a CBS.

The concept of a CBS helps to identify those situations where a bias may be observed in the frequency with which different pheromone values are updated by solutions produced by an ACO algorithm, without requiring full knowledge of the distribution of solution characteristics in the construction tree. Furthermore, the existence of an underlying construction bias—a feature which is typically easily identified from knowledge of the problem and the constructive algorithm used—can indicate that no pheromone model can be a CBS for a particular problem. However, neither of these can necessarily be used to predict exactly *how* a particular pheromone will be biased.

5. Predicting bias in pheromone models

Given that any pheromone model for a problem with a constructed solution bias will also exhibit a bias, it is important to be able to predict the effects of that bias in order to choose the best model. However, it is only with specialised insight into the nature of a problem and of the likely frequency of solution characteristics that estimates can be made of the number and kind of sequences each characteristic will appear in. The JSP is one problem in which predictable relationships do exist, and the literature on that problem is discussed first, together with some new findings. Novel techniques are then developed for the study of bias in other problems and pheromone models.

The three pheromone models discussed with respect to the JSP are $\mathfrak{C} \times \mathfrak{C}$ (denoted by $\mathsf{PH}_{\mathsf{suc}}$), $\mathfrak{C} \times P$ (denoted by $\mathsf{PH}_{\mathsf{pos}}$) and the second order pheromone developed by Blum and Sampels [2] that models the relative order of operations that use the same machine (denoted by $\mathsf{PH}_{\mathsf{rel}}$). Blum and Sampels [2,5] found that $\mathsf{PH}_{\mathsf{suc}}$ performs poorly on this problem. Further investigation revealed that sequences corresponding to poorer solutions were typically characterised by having runs of operations from the same job. In terms of CBSs, solution characteristics from $\mathsf{PH}_{\mathsf{suc}}$ that correspond to selecting an operation from the same job as that last selected occur in more sequences than those which do not have this property. In contrast, in the best performing pheromone for this problem, $\mathsf{PH}_{\mathsf{rel}}$, solution characteristics associated with high quality solutions appear more frequently than those associated with poor solutions. Empirical results do not suggest that $\mathsf{PH}_{\mathsf{pos}}$ is strongly biased in either direction.

Montgomery, Randall and Hendtlass [9] expand on these findings, showing that the JSP has a *construction bias* that favours poor solutions and a *representation bias* that favours good solutions. The underlying causes of these biases are that selecting an operation from the same job as that last selected necessarily restricts later choices (introducing a construction bias in favour of such sequences) *and* makes it more likely that, later in construction, two or more operations from the same job will be placed in sequence. At the same time, such sequences can be perturbed very little before the solution they represent changes. Sequences in which sequential operations are rarely from the same job may be perturbed to a greater extent yet still represent the same solution. These underlying biases are the direct causes of the biases observed in $\mathsf{PH}_{\mathsf{suc}}$ and $\mathsf{PH}_{\mathsf{rel}}$, because of the characteristics they model.

These relationships suggest that patterns may be observed between the number of sequences in which each solution characteristic appears and the mean cost or value of those solutions, which could become a relatively simple test for predictable patterns of bias in other problems. Fig. 1 describes a nine operation, three job, three machine JSP instance, described by Montgomery et al. [9]. Fig. 2 shows plots of the number of sequences in which each solution characteristic appears against the mean cost of the solutions represented by those sequences, for each of $\mathsf{PH}_{\mathsf{suc}}$, $\mathsf{PH}_{\mathsf{pos}}$ and $\mathsf{PH}_{\mathsf{rel}}$ applied to that instance. The mean cost of all sequences is shown as a dashed line. Although the three plots show some commonality in structure, key differences are also apparent. At the extremities of frequency, the distribution for $\mathsf{PH}_{\mathsf{suc}}$ shows a range of solution costs associated with solution characteristics, while $\mathsf{PH}_{\mathsf{pos}}$ and $\mathsf{PH}_{\mathsf{rel}}$ have much tighter ranges. That is, $\mathsf{PH}_{\mathsf{pos}}$ and $\mathsf{PH}_{\mathsf{rel}}$ show a stronger relationship between frequency of usage and associated solution cost. Moreover, those solution characteristics that correspond to the best solutions have a frequency of less than half

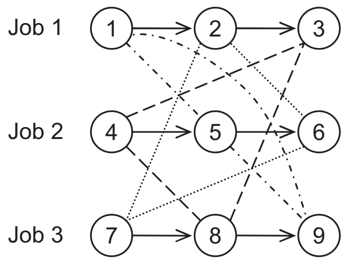

Fig. 1. Nine operation, three job, three machine JSP instance. The processing order of operations within each job is shown by directed arcs. Operations that require the same machine are joined with dashed arcs. The processing times of operations are $p(1)=p(5)=p(9)=10$, $p(2)=p(6)=p(7)=20$, $p(3)=p(4)=p(8)=30$, where $p(i)$ is the processing time of operation i.

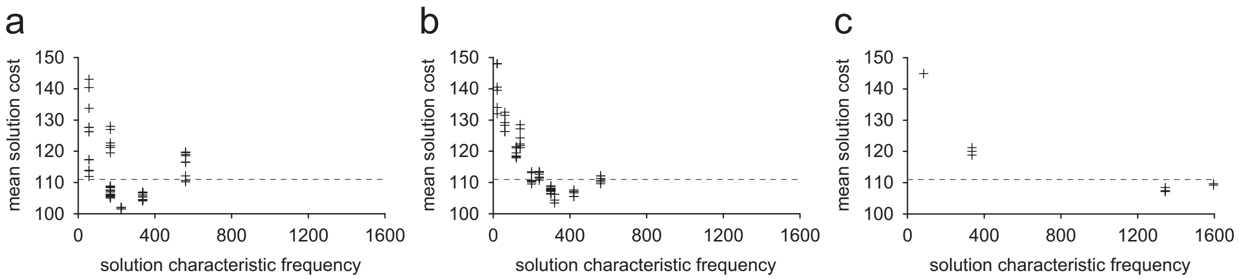

Fig. 2. Frequency of usage of solution characteristics against the mean cost of solutions they describe for a nine operation JSP instance [9], for each of (a) PH_{suc}, (b) PH_{pos} and (c) PH_{rel}. The mean cost of all sequences is shown as a dashed line.

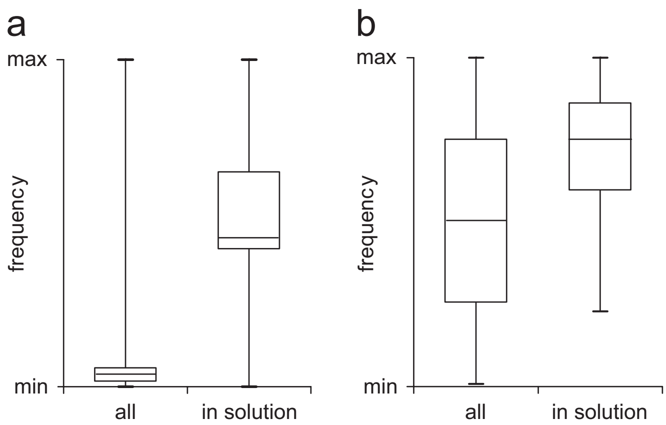

Fig. 3. Distributions of solution characteristic frequencies for all solution characteristics and for those to which ACO converges using (a) PH_{suc} and (b) PH_{rel}.

the maximum in PH_{suc}, just over half the maximum in PH_{pos} and 84% of the maximum in PH_{rel}. The most frequently occurring solution characteristics in PH_{rel} are also associated with good solutions. Similar results were obtained for other small instances.

Analysis was subsequently made of the pheromone values in PH_{suc} and PH_{rel} models in an ACO algorithm applied to various JSP instances. When applied to small problems, a clear relationship was observed between the number of solution sequences in which each solution characteristic appears and its likelihood of being reinforced. This relationship is also evident in non-trivial instances, where the frequency of solution characteristics was estimated by sampling 10^8 randomly generated sequences. Fig. 3(a) and (b) show box plots that compare the distributions of estimated frequencies of all solution characteristics against those to which the algorithm converged on the `ft10` instance, for PH_{suc} and PH_{rel}, respectively. The average quality of solutions produced that contain high frequency solution characteristics is considerably better when using PH_{rel} than PH_{suc}, which accords with the findings described above.

These same techniques were used to examine pheromone models for the MKP and GAP. Both studies begin by looking for a predictable pattern in solution characteristic frequency, before comparing solution characteristic frequency against the value of solutions each appears in and, finally, against observed pheromone values in an ACO algorithm.

5.1. MKP

The instances studied come from the `mknap1` data set, available at the OR-Library [19]. This data set contains seven instances of different sizes. Here, each instance is referred to as `mknap1-`*n*`item`, where *n* is the size of the problem.

In the MKP, it is expected that there exists an inverse relationship between an item's resource requirements (often referred to as *weight*) and its frequency of inclusion in solutions. Indeed, a linear regression of the data from the

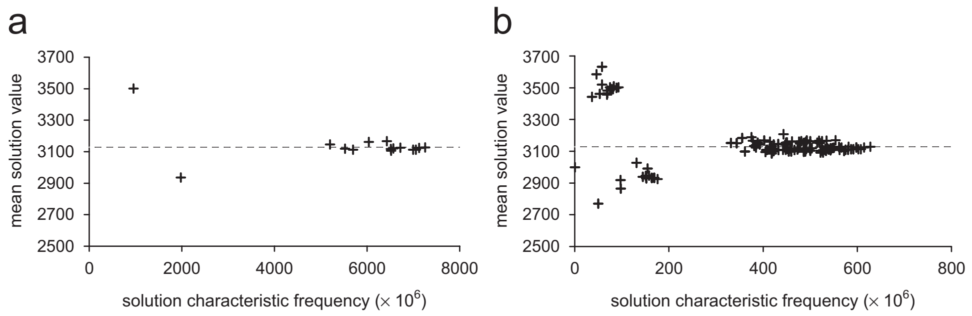

Fig. 4. Potential bias interaction in the mknap1-15item MKP instance. (a) and (b) Frequency of use of solution characteristics against mean value of solutions they describe from $\mathsf{PH}_{\mathsf{items}}$ and $\mathsf{PH}_{\mathsf{suc}}$, respectively. The mean value of all sequences is shown as a dashed line.

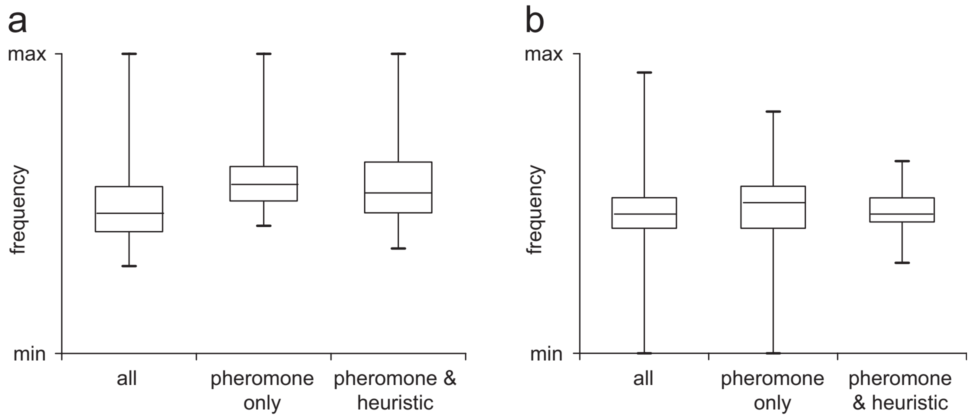

Fig. 5. Distributions of solution characteristic frequencies for all solution characteristics and those to which ACO converges using $\mathsf{PH}_{\mathsf{items}}$ (a) and $\mathsf{PH}_{\mathsf{suc}}$ (b) without and with a heuristic bias for a 100 items MKP instance.

mknap1-6item, mknap1-10item and mknap1-15item instances reveals strong negative correlations of $\rho = -0.91$, -0.94 and -0.97, respectively. Analyses of the pheromone models $\mathfrak{C}$ (denoted $\mathsf{PH}_{\mathsf{items}}$) and $\mathsf{PH}_{\mathsf{suc}}$ (i.e., modelling successively chosen items) confirms strong negative correlations ($\rho = -0.95$ and -0.96, respectively) between solution characteristic frequency and the resource requirements of the item(s) included by each characteristic.

In the MKP, item weight and value may be related. The correlations between the two for the mknap1-6item, mknap1-10item and mknap1-15item instances are $\rho = 0.74$, 0.91 and 0.83, respectively. Consequently, it may be expected that solution characteristics associated with high value solutions would also appear in fewer sequences and be reinforced less frequently. Fig. 4(a) and (b) plot the frequency of solution characteristic usage against the mean value of solutions represented for the mknap1-15item instance for $\mathsf{PH}_{\mathsf{items}}$ and $\mathsf{PH}_{\mathsf{suc}}$. The figure shows three groups of solutions, with the bulk of solution characteristics clustered around the mean value of sequences. Further examination reveals that, apart from one group of solutions that includes a heavy and highly valuable item and another that includes a heavy and moderately valuable item, the majority of solution characteristics appear in so many solutions (with a correspondingly large range of values) that any bias towards one characteristic may not necessarily bias the search towards good or bad solutions.

Observations of ACO's performance on the MKP show that both pheromone models can produce good solutions, significantly better than $\mathrm{ACO}_{\mathrm{undir}}$. However, it may still be the case that the algorithm converges to solutions containing those characteristics that appear most frequently, rather than those that are part of the best solution(s). Fig. 5(a) shows box plots that compare the distributions of estimated frequencies of all solution characteristics against those with relatively high pheromone values for a 100 item instance in the mknapcb1 problem set. The plots show that more frequently occurring solution characteristics are reinforced to a greater extent. However, it should be noted that the

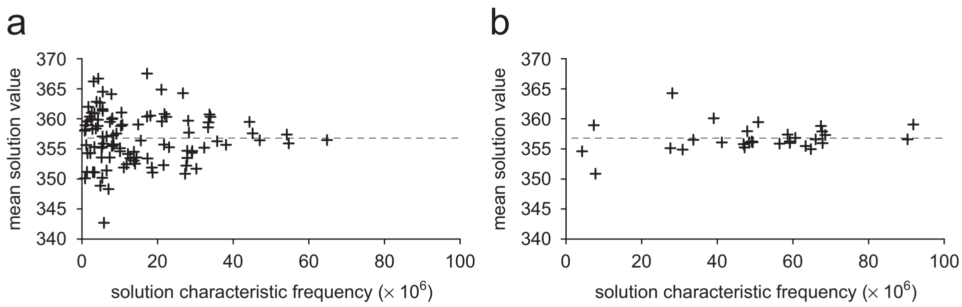

Fig. 6. Potential bias interaction in the `gap2-1` GAP instance. (a) and (b) Frequency of use of solution characteristics from $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$, respectively, against mean value of solutions they describe. The mean value of all sequences is shown as a dashed line.

algorithm finds solutions close to the best known. A heuristic bias that favours high value items (which implicitly favours those characteristics that appear less frequently) improves the algorithm's performance.

Using $\mathsf{PH}_{\mathsf{suc}}$, the algorithm quickly converges to a solution. Fig. 5(b) shows box plots comparing the distributions of solution characteristic frequency for the same groups as in (a). These plots suggest that $\mathsf{PH}_{\mathsf{suc}}$ is affected to a lesser extent by solution characteristic frequency, although it typically produces poorer solutions than $\mathsf{PH}_{\mathsf{items}}$.

5.2. GAP

The instances discussed in this section are taken from the `gap1`, `gap2` and `gap12` data sets, available at the OR-Library [19]. Each set contains a number of instances, which are labelled here according to `gap`*k*-*n*, where *k* is the data set number and $n \in [1, 5]$ is the instance from that data set.

As with items in the MKP, assignments with relatively high capacity utilisation would be expected to occur less frequently in the solutions that may be produced. Linear regression of the proportion of an agent's capacity used by an assignment against that assignment's frequency of use, for `gap1-1` and `gap2-1` instances, shows moderate correlations of $\rho = -0.73$ and -0.79, respectively. The two pheromone models considered with this problem are $\mathfrak{C}_{\mathrm{it}} \times \mathfrak{C}_{\mathrm{res}}$ (denoted $\mathsf{PH}_{\mathsf{assigns}}$) and $\mathfrak{C}_{\mathrm{res}} \times \mathfrak{C}_{\mathrm{res}}$ (i.e., successive agents in the solution sequence, denoted $\mathsf{PH}_{\mathsf{suc-res}}$).

Unlike the MKP, high resource requirements are not correlated with high value in the instances examined.[6] Thus, solution characteristics are unlikely to be strongly associated with solutions of a particular value. Fig. 6(a) and (b) plot the frequency of solution characteristic usage against the mean value of solutions represented for the `gap2-1` instance for $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$. Although some solution characteristics with a low frequency of use are clearly associated more strongly with either high or low value solutions, as a solution characteristic is used in more solutions, the mean value of those solutions approaches the mean value of all solutions.

Observations of ACO's performance on the GAP show that both pheromone models can produce better solutions than $\mathrm{ACO}_{\mathrm{undir}}$. However, the performance of $\mathsf{PH}_{\mathsf{suc-res}}$ is significantly worse than $\mathsf{PH}_{\mathsf{assigns}}$ and not significantly better than $\mathrm{ACO}_{\mathrm{undir}}$. This does not appear to be because of any particular bias towards frequently used characteristics but because the model is simply inappropriate for this kind of problem. Analyses of final pheromone values from $\mathsf{PH}_{\mathsf{assigns}}$ used with the `gap2-1` and `gap2-2` instances show that the frequencies of those solution characteristics to which the algorithm converged are typically higher than the underlying distribution of solution characteristic frequencies. However, the same is true of those found in the optimal solutions for these instances, suggesting that the algorithm is, in fact, learning the best solution characteristics. Given these problems are tightly constrained, it is plausible that the best feasible solutions will contain a number of assignments that make little use of agent resources, and which accordingly appear in a larger number of solutions. Fig. 7 shows box plots that compare the distributions of frequencies of all solution characteristics against those to which the algorithm converged and those belonging to the three optimal solutions to the `gap2-1` instance.

A similar result was observed in the application of $\mathsf{PH}_{\mathsf{assigns}}$ to the `gap12-1` instance. Fig. 8 shows box plots that compare the distributions of estimated frequencies of all solution characteristics against those to which the algorithm

[6] While many GAPs are minimisation problems, the `gap1` and `gap2` problem sets contain maximisation problems.

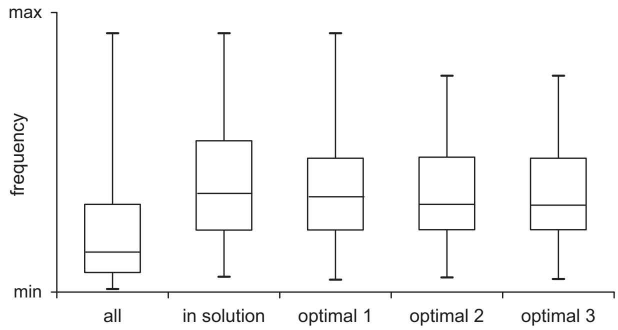

Fig. 7. Distributions of solution characteristic frequencies for all solution characteristics, those to which ACO converges using $\mathsf{PH}_{\mathsf{assigns}}$ and the three optimal solutions to the `gap2-1` instance.

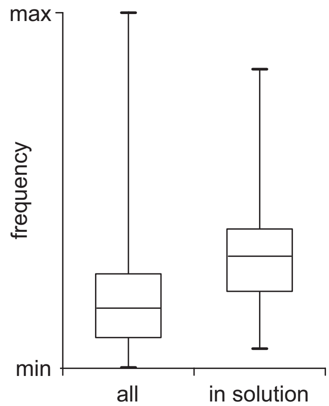

Fig. 8. Distributions of solution characteristic frequencies for all solution characteristics and those to which ACO converges using $\mathsf{PH}_{\mathsf{assigns}}$ with the `gap12-1` instance.

converged.[7] Although the algorithm converges on frequently occurring solution characteristics, the mean value of solutions containing those characteristics is high and solutions produced by the algorithm are often within 5% of the optimal for this instance.

Unlike the MKP, introducing a heuristic bias in favour of less frequent assignments is unlikely to improve solutions. As feasible solutions are so rare in these problems, a heuristic bias that favours assignments that make little use of resources, and hence which appear more frequently, helps produce better solutions [7].

6. Selecting pheromone models considering bias

Ideally, the bias different pheromone models exhibit should be identified in order to select the model that will be most effective. However, this task is non-trivial and, indeed, may be impossible for many problems. The JSP is a problem that does have a predictable bias, while other problems, such as the MKP and GAP examined above, have predictable patterns in the frequency with which certain solution characteristics will be updated, yet no predictable bias towards particular kinds of solutions.

Given a problem with no constructed solution biases, any pheromone model that may be applied to that problem will be a CBS [7]. However, it does not follow that every applicable pheromone model will produce equivalent performance. Montgomery et al. [10] suggest that pheromone models be chosen that model solution characteristics

[7] Frequencies were estimated by sampling 10^8 randomly generated sequences.

that directly contribute to solution cost and which consequently are identity oriented. Certainly, results from the small number of comparative studies of alternative pheromones that have been undertaken lend support to this claim [20,21]. Based on this premise they devise a decision algorithm to select pheromone models, which was used to predict the best performing models tested in the study presented in Section 7 below.

If a problem has a constructed solution bias, then under all practical conditions there does not exist a pheromone model that will be free from bias. Specifically, if a problem has a construction bias then there does not exist a pheromone model that is a CBS, and thus an identity-oriented pheromone will not be free from bias [7]. Furthermore, given what is currently known about pheromone models with this property, the use of such a pheromone does not entail a reduction in bias. For example, $\mathsf{PH}_{\mathsf{rel}}$ is an identity-oriented pheromone model for the JSP which exhibits a strong bias [7,9]. Fortunately, given the commonality in structure of many JSP instances, this bias typically favours good solutions. However, Blum [13] describes contrived instances where it performs very badly.

Blum [13] investigates two ways to counteract undesirable bias effects in ACO algorithms by altering the way pheromone values are updated. The first is to use the iteration best solution to update pheromone values, as is done in some ant colony system (ACS) [22] and $\mathcal{MAX}$–$\mathcal{MIN}$ ant system ($\mathcal{MM}$AS) [23] algorithms, rather than allow all ants to update solutions. While this improves the performance of pheromone models with a known, unfavourable bias, other models are still superior. The other technique is the use of stochastic gradient ascent (SGA) (see, e.g., [24]), which is possible given the relationship between SGA and ACO identified by Meuleau and Dorigo [25]. The technique was used in an ACO algorithm for the JSP using $\mathsf{PH}_{\mathsf{suc}}$, which has a known, unfavourable bias, producing results comparable with the best performing pheromone model for that problem. Despite the improved performance achieved by using the SGA update, a number of disadvantages were identified, leading to the recommendation that a standard ACO pheromone update should be used in conjunction with an appropriate pheromone model.

If the nature of the bias in a pheromone model is unknown or cannot be determined using available analysis techniques, then modelling solution characteristics related to cost offers the following advantages over representation-oriented alternatives. First, as solutions are represented uniquely, the algorithm will learn only one estimate of a solution's quality. A representation-oriented model can learn different estimates of a solution's quality, one for each of its representations. Second, in an identity-oriented model there is a clear relationship between the solution characteristics modelled and the impact of their inclusion on solution cost. Both of these suggest that such pheromone models should learn more effectively which solution characteristics to combine to produce good solutions.

If the nature of the biases in a collection of alternative pheromone models *is* known, then this knowledge may be used to select the one with the most favourable bias, even if it is not identity-oriented. Thus, it is by considering the existence and, subsequently, nature of biases in pheromone models that informed decisions concerning the most appropriate pheromone model may be made.

7. Pheromone model comparative performance

This section summarises the results of a large number of tests of the performance of ACO with a range of pheromone models across a number of problems. An extended discussion of these results, with details of the instances studied and tables of numerical results, is available in Montgomery [7]. The purpose of these experiments is to test the hypothesis that identity-oriented pheromones, in particular those that model solution characteristics that directly contribute to solution cost, perform best. Given that this aim relates to the relative performance of alternative pheromone models given otherwise identical ACO algorithms, comparisons are not made between the performance of the algorithms described here and those described in the literature.

Six types of *CO Problem* were studied, the TSP, MKP, group-shop scheduling problem (GSP, a generalisation of the JSP and OSP), QAP, GAP and car sequencing problem (CSP) [26]. These problems were chosen as they are well-studied in the literature, exhibit a range of constructed solution biases and are amenable to solution using a range of different pheromone models.

The decision algorithm described by Montgomery et al. [10] has been used to predict which pheromone models should perform best. Alternative models were selected either from those used in the literature or from those that are potentially applicable.

In order that differences in the relative performance of alternative pheromones be established independently of the precise way in which pheromone information is updated, two alternative ACO algorithms were studied: a modified

Table 2
Basic pheromone models

Problem entities	Pheromone	Name
$\mathfrak{C}$	$C = \mathfrak{C}$	$\mathsf{PH}_{\mathsf{items}}$
$\mathfrak{C}$	$C \subset \mathfrak{C} \times \mathfrak{C}$ (successive components)	$\mathsf{PH}_{\mathsf{suc}}$
$\mathfrak{C}$	$C \subset \mathfrak{C} \times \mathfrak{C}$ (copresent components)	$\mathsf{PH}_{\mathsf{pairs}}$
$\mathfrak{C}$, P	$C \subset \mathfrak{C} \times P$	$\mathsf{PH}_{\mathsf{pos}}$
$\mathfrak{C}_{\mathrm{it}}, \mathfrak{C}_{\mathrm{res}}$	$C = \mathfrak{C}_{\mathrm{it}} \times \mathfrak{C}_{\mathrm{res}}$	$\mathsf{PH}_{\mathsf{assigns}}$
$\mathfrak{C}_{\mathrm{it}}, \mathfrak{C}_{\mathrm{res}}$	$C \subset \mathfrak{C}_{\mathrm{res}} \times \mathfrak{C}_{\mathrm{res}}$ (successive resources)	$\mathsf{PH}_{\mathsf{suc\text{-}res}}$

version of ACS [22] and a standard $\mathcal{MM}$AS [23].[8] Some of the control parameters were varied across the six problems, using initial testing to determine good values.[9] The values used for a particular problem are the same as those used by Montgomery [7]. All runs used 10 ants over 3000 iterations.

Algorithms for the TSP, MKP, GSP and GAP were implemented with the capability to use a heuristic bias, the details of which are given in their respective sections. Algorithms for the TSP and QAP were implemented with the capability to apply a local search procedure to solutions produced, details of which are given by Montgomery [7].

Although a number of different pheromone models are studied in this section, many have either been introduced above or are used across several problems. These "basic" models are described in Table 2, with other models introduced as needed later. A summary of all pheromone models used with each problem appears in Table 10 at the end of this section.

The performances of alternative pheromone models for each problem are compared against each other as well as against $\mathrm{ACO}_{\mathrm{undir}}$, which is identical except for not using pheromone information. When a heuristic bias is used with $\mathrm{ACO}_{\mathrm{undir}}$, the resulting algorithm is referred to as $\mathrm{ACO}_{\mathrm{heur}}$.

7.1. Variables and analysis techniques used

Each combination of pheromone model (including no pheromone, $\mathrm{ACO}_{\mathrm{undir}}$) and the use or not of a heuristic bias or local search (where supported) was tested across 10 random seeds. The output from each experiment is a frequency distribution of the costs (or values) of solutions produced during the algorithm's run, expressed as the relative percentage deviation (RPD) from the optimal solution's cost or value. RPD is calculated according to

$$RPD = \frac{|C - C_{\mathrm{best}}|}{C_{\mathrm{best}}}, \tag{1}$$

where C is the solution's cost/value and C_{best} is the optimal (or best known) solution cost/value for the problem in question. As the distributions from each run are non-normal, the data were summarised using the minimum and median RPD. Comparisons between different pheromone models were conducted using non-parametric statistical tests on these two measures. Additionally, as the GAP is a highly constrained problem in which infeasible solutions may be produced, pheromone models were also compared on the proportion of feasible solutions produced, using the same statistical tests.

Comparisons between alternative pheromones (and $\mathrm{ACO}_{\mathrm{undir}}$) are presented in tabular form as shown in Table 3. This is a contrived example comparing the minimum RPD results of $\mathrm{ACO}_{\mathrm{undir}}$ with PH_{a} and PH_{b}, two imaginary models, and is read as follows. To determine how the performance of $\mathrm{ACO}_{\mathrm{undir}}$ compares against PH_{a}, locate $\mathrm{ACO}_{\mathrm{undir}}$ in the left-most column and then within that row locate the cell in the column under PH_{a}. In this instance, the $>$ symbol indicates that $\mathrm{ACO}_{\mathrm{undir}}$ produced higher values for minimum RPD than PH_{a}. The "($\alpha = 1\%$)" below indicates that this result is statistically significant at the 1% level. It does not indicate the magnitude of the difference. If a difference exists between two alternatives but is not statistically significant, then no significance level is shown below the direction indicator.

[8] Initial testing suggested that the greedy bias of ACS, controlled by the q_0 parameter, often leads to inferior performance than when it is not used. Consequently, this parameter was set to zero in all ACS experiments.

[9] The values used typically produced the best performance across all competing pheromone models.

Table 3
Example comparison table

	$\mathsf{PH_a}$	$\mathsf{PH_b}$
$\mathrm{ACO_{undir}}$	$>$ ($\alpha = 1\%$)	$<$ ($\alpha = 10\%$)
$\mathsf{PH_a}$		$<$

The direction of any difference is shown with the significance level below if the result is statistically significant based on a Mann–Whitney test. For example, the relative performance of $\mathsf{PH_a}$ and $\mathsf{PH_b}$ is read by locating $\mathsf{PH_a}$ in the left-most column and then taking the difference direction indicator, in this case $<$, from the column corresponding to $\mathsf{PH_b}$, indicating the result $\mathsf{PH_a} < \mathsf{PH_b}$, which in this case is not statistically significant.

Table 4
Pairwise comparisons of minimum RPD achieved by $\mathrm{ACO_{undir}}$ and ACO with different pheromone models for the TSP

	$\mathsf{PH_{suc}}$	$\mathsf{PH_{pos}}$
$\mathrm{ACO_{undir}}$	$>$ ($\alpha = 1\%$)	$>$ ($\alpha = 1\%$)
$\mathsf{PH_{suc}}$		$<$ ($\alpha = 1\%$)

7.2. TSP

The TSP instances studied were taken from the TSPLIB [27] and range in size from 24 to 532 cities. The heuristic bias used is the inverse of the distance between two cities, which consequently favours shorter edges. As the local search procedure is computationally very costly it was only used with instances up to size 100.

The two pheromones used with the TSP were $\mathsf{PH_{suc}}$ and $\mathsf{PH_{pos}}$, with the former being most strongly associated with solution cost. Table 4 shows the results of Mann–Whitney tests comparing pairs of algorithms. Although both minimum and median RPD were compared, the results are almost entirely identical and hence only one table of comparisons is shown. The order of performance of the alternatives was fairly constant across ACS and $\mathscr{M}\mathscr{M}$AS, as well as when heuristic bias and local search are used, with the chief exception being that when neither heuristic bias nor local search is used, the difference between $\mathsf{PH_{suc}}$ and $\mathsf{PH_{pos}}$ is not statistically significant, although the direction of difference is the same.

Overall, $\mathrm{ACO_{undir}}$ (here representing both itself and its heuristically guided counterpart $\mathrm{ACO_{heur}}$), $\mathsf{PH_{suc}}$ and $\mathsf{PH_{pos}}$ may be ranked in terms of solution cost as

$$\mathsf{PH_{suc}} \prec \mathsf{PH_{pos}} \prec \mathrm{ACO_{undir}},$$

where $A \prec B$ means that, in general, A produces better solutions than B.

7.3. MKP

The MKP instances studied belong to the `mknap1` and `mknapcb1` data sets, available at the OR-Library [19]. All instances from the `mknap1` set, ranging in size from 6 to 50 items, were studied, while six instances from the `mknapcb1` set, all of size 100, were used.

Solutions to this problem are constructed as sequences of the items chosen, which supports a number of alternative pheromone models. The four examined were $\mathsf{PH_{items}}$, $\mathsf{PH_{pairs}}$, $\mathsf{PH_{suc}}$ and $\mathsf{PH_{pos}}$. The last two are representation-oriented models, while the second is a higher order version of $\mathsf{PH_{items}}$. As $\mathsf{PH_{pairs}}$ can only inform decisions once a solution contains at least one component, when this model was used the initial component was selected randomly, to encourage exploration. The potentially multiple pheromone values from $\mathsf{PH_{pairs}}$ are combined by taking the mean for the relevant solution characteristics. The heuristic bias used is an item's profit.

Table 5
Pairwise comparisons of minimum RPD achieved by $\text{ACO}_{\text{undir}}$ and $\mathcal{MM}$AS with different pheromone models for the MKP, using a heuristic bias

	PH_{items}	PH_{pairs}	PH_{suc}	PH_{pos}
ACO_{heur}	$>$ $(\alpha = 1\%)$	$>$ $(\alpha = 1\%)$	$>$ $(\alpha = 1\%)$	$>$ $(\alpha = 1\%)$
PH_{items}		$<$	$<$ $(\alpha = 10\%)$	$>$ $(\alpha = 5\%)$
PH_{pairs}			$<$ $(\alpha = 5\%)$	$>$ $(\alpha = 1\%)$
PH_{suc}				$>$ $(\alpha = 1\%)$

Across the four combinations of ACO algorithm (ACS or $\mathcal{MM}$AS) and heuristic versus no heuristic bias the following relationships were found to be consistent:

- ACO with any pheromone outperforms $\text{ACO}_{\text{undir}}$ and ACO_{heur}.
- All other pheromones outperform PH_{suc}.
- PH_{items} pheromone outperforms PH_{pairs}.

The performance of all algorithms was improved by the use of the heuristic bias.

However, the results also showed some sensitivity to the ACO algorithm used and to whether or not the heuristic bias was used. Table 5 summarises the comparisons of minimum RPD for $\mathcal{MM}$AS using the heuristic bias, which yielded the best performance for all models. The key differences between the results for $\mathcal{MM}$AS and ACS were that the difference between PH_{items} and PH_{pairs} was statistically significant and, most notably, that PH_{items} and PH_{pairs} produced better results than PH_{pos}. However, the best results were achieved using PH_{pos} under $\mathcal{MM}$AS with the heuristic bias. In these experiments the following ranking of alternative pheromones and $\text{ACO}_{\text{undir}}$ (or ACO_{heur}) in terms of solution value appears to hold when either the heuristic bias is not used or the ACO algorithm is $\mathcal{MM}$AS:

$$\text{PH}_{\text{pos}} \succ \text{PH}_{\text{items}} \succ \text{PH}_{\text{pairs}} \succ \text{PH}_{\text{suc}} \succ \text{ACO}_{\text{undir}}$$

while when ACS is used with the heuristic bias:

$$\text{PH}_{\text{items}} \succ \text{PH}_{\text{pairs}} \succ \text{PH}_{\text{pos}} \succ \text{PH}_{\text{suc}} \succ \text{ACO}_{\text{undir}}.$$

The often superior performance of PH_{pos}, a pheromone that has never been considered for use with the MKP and which intuitively appears inappropriate, requires further investigation.

7.4. GSP

The GSP described by Blum and Sampels [5] is a generalisation of the JSP and OSP in which operations within each job are partitioned into *groups*, with precedence constraints imposed between groups. In the JSP, each operation forms its own group, while in the OSP all operations within a job form a single group. By altering the size of groups a range of problems intermediate between the two problem types may be created.

Nine GSP instances were studied, ranging in size from 100 to 225 operations, taken from a data set used by Blum [13]. These nine instances are based on three original problems, two JSP instances and one true GSP instance, with each problem altered to form a JSP, GSP and OSP instance.

The pheromone models used with the GSP were PH_{suc}, PH_{pos} and PH_{rel}. The first two are representation-oriented while PH_{rel} is identity-oriented. The heuristic bias used is the inverse of an operation's processing time.

Considering all problem instances, PH_{rel} clearly outperforms all other approaches, regardless of the ACO algorithm used and whether or not the heuristic bias is used. PH_{pos} is next best, and produces almost equivalent results when OSP instances are considered separately. Table 6 shows the results of Mann–Whitney tests comparing alternative approaches when the heuristic bias is not used. These results were the same for ACS and $\mathcal{MM}$AS.

Table 6
Pairwise comparisons of minimum RPD achieved by ACO_{undir} and ACO using different pheromone models for the GSP when the heuristic bias is not used

	PH_{rel}	PH_{suc}	PH_{pos}
ACO_{undir}	$>$ ($\alpha = 1\%$)	$<$	$>$ ($\alpha = 1\%$)
PH_{rel}		$<$ ($\alpha = 1\%$)	$<$ ($\alpha = 1\%$)
PH_{suc}			$>$ ($\alpha = 1\%$)

Table 7
Pairwise comparisons of minimum RPD achieved by ACO_{undir} and ACO with different pheromone models for the QAP, under ACS or $\mathcal{MM}$AS when local search is not used

	$PH_{assigns}$	$PH_{suc\text{-}res}$
ACO_{undir}	$>$ ($\alpha = 1\%$)	$>$ (α=1–10%)
$PH_{assigns}$		$<$ ($\alpha = 1\%$)

Considering the JSP, GSP and OSP instances as a whole, or within JSP and GSP instances, the following ranking of alternative pheromones and ACO_{undir} (or ACO_{heur}) holds:

$$PH_{rel} \prec PH_{pos} \prec ACO_{undir} \prec PH_{suc}.$$

On OSP instances, the relative order PH_{suc} and ACO_{undir} (and ACO_{heur}) is reversed.

7.5. QAP

The QAP instances studied were taken from the QAPLIB [28], and range in size from 12 to 64. As the local search procedure used with this problem is computationally very costly it was only used with instances up to size 35.

The two pheromone models used with the QAP were $PH_{assigns}$ and $PH_{suc\text{-}res}$. The first is identity-oriented while the second is representation-oriented. As there is a choice concerning the order in which facilities are assigned in this problem, three alternatives were examined: the order they appear in the instance description; a dynamic, randomised order; and non-increasing order of their total flow requirements. Comparisons between the two pheromones were carried out only within results for the two static assignment orders, as $PH_{suc\text{-}res}$ requires a static assignment order to be sensibly applied.

Table 7 shows the results of Mann–Whitney tests comparing alternative approaches when local search is not used. These results vary little between ACS and $\mathcal{MM}$AS, or between assignment orders, although some degree of variability was found in the significance level of the difference between ACO_{undir} and $PH_{suc\text{-}res}$. Using the randomised assignment order, $PH_{assigns}$ consistently produces better solutions than ACO_{undir}. Thus, when local search is not used, ACO_{undir}, $PH_{assigns}$ and $PH_{suc\text{-}res}$ may be ranked in terms of solution cost as

$$PH_{assigns} \prec PH_{suc\text{-}res} \prec ACO_{undir}.$$

When local search is used, ACO_{undir}, $PH_{assigns}$ and $PH_{suc\text{-}res}$ perform similarly. However, on the two largest instances with which local search was used, $PH_{assigns}$ found better solutions more often than ACO_{undir} and $PH_{suc\text{-}res}$. Furthermore, comparing the median RPD results for the three reveals that $PH_{assigns}$ finds proportionally more good quality solutions than $PH_{suc\text{-}res}$, which in turn produces proportionally more good quality solutions than ACO_{undir}. The ability of $PH_{assigns}$ to converge on good quality solutions when local search is used suggests that on larger instances it may perform better than ACO_{undir} with local search.

Table 8
Pairwise comparisons of pheromone models for the GAP when using a static assignment order when the heuristic bias is not used

	Minimum RPD		Feasible solutions	
	$\mathsf{PH}_{\mathsf{assigns}}$	$\mathsf{PH}_{\mathsf{suc-res}}$	$\mathsf{PH}_{\mathsf{assigns}}$	$\mathsf{PH}_{\mathsf{suc-res}}$
$\mathrm{ACO}_{\mathrm{undir}}$	$>$ ($\alpha = 1\%$)	$>$	$<$ ($\alpha = 1\%$)	$<$ ($\alpha = 1\%$)
$\mathsf{PH}_{\mathsf{assigns}}$		$<$ ($\alpha = 1\%$)		$>$ ($\alpha = 1\%$)

7.6. GAP

The GAP instances studied were taken from the `gap1` through `gap12` sets, available at the OR-Library [19], and range in size from 15 tasks, five agents to 60 tasks, 10 agents. As with the QAP, there exists a choice over the order in which tasks are assigned. In addition to assigning items in the order they appear in the instance description and assigning items in a dynamic, randomised order, a number of static and dynamic heuristic assignment orders designed to increase the likelihood of producing feasible solutions were also used (see [7] for details). The heuristic bias used is the inverse of an agent's capacity used by a candidate assignment, which consequently favours low resource utilisation and so increases the likelihood of producing feasible solutions.

The two pheromone models used were $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$. Table 8 shows the results of Mann–Whitney tests comparing alternative approaches when the heuristic bias is not used. These comparisons include both performance in terms of solution value and the proportion of feasible solutions produced. These results were the same for the two static assignment orders considered and when using ACS or $\mathscr{M}\mathscr{M}$AS. The only exceptions were found when using the heuristic bias, where $\mathrm{ACO}_{\mathrm{heur}}$ performed equivalently to $\mathsf{PH}_{\mathsf{suc-res}}$ under $\mathscr{M}\mathscr{M}$AS in terms of solution value, while the significance level of the difference in solution value between $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$ is 5%. Results for comparisons involving median RPD were the same, with the exception that the difference between $\mathrm{ACO}_{\mathrm{undir}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$ was found to be statistically significant when using the heuristic bias. Better results were obtained with all algorithms when the heuristic bias was used, which is to be expected given it leads to a larger number of feasible solutions being produced and consequently to greater exploration of feasible solution space.

Comparisons between $\mathrm{ACO}_{\mathrm{undir}}$ and $\mathsf{PH}_{\mathsf{assigns}}$ with the dynamic assignment orders found that $\mathsf{PH}_{\mathsf{assigns}}$ produces better solutions and more feasible solutions than $\mathrm{ACO}_{\mathrm{undir}}$ regardless of the assignment order, ACO algorithm or whether the heuristic bias is used or not. All results are statistically significant at the 1% level.

Across assignment orders, $\mathrm{ACO}_{\mathrm{undir}}$ (also representing $\mathrm{ACO}_{\mathrm{heur}}$), $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$ may be ranked in terms of solution value as

$$\mathsf{PH}_{\mathsf{assigns}} \succ \mathsf{PH}_{\mathsf{suc-res}} \succ \mathrm{ACO}_{\mathrm{undir}}.$$

In terms of the proportion of feasible solutions produced, the three alternatives may be ranked in the same order, where $A \succ B$ is interpreted as A produces more feasible solutions than B. This last result is to be expected given that both $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{suc-res}}$ are only reinforced by feasible solutions, and so are more likely to make assignments that will lead to more feasible solutions being produced.

The best performing heuristic assignment orders in terms of solution value were the worst in terms of the proportion of feasible solutions produced, and vice versa, although assigning items in the order in which they appear in the instance description led to the worst results on both measures. The approximately inverse relationship between the proportion of feasible solutions produced and solution quality (when using the heuristic assignment orders) suggests that the best solutions to these instances are found near the bounds of feasible space. Hence, there can be a trade-off between seeking high quality solutions and keeping the probability of producing feasible solutions high [29].

Table 9
Pairwise comparisons of pheromone models for the CSP when using $\mathcal{MM}$AS using a dynamic, randomised assignment order

	$\mathsf{PH}_{\text{assign-pairs}}$	$\mathsf{PH}_{\text{assigns}}$	$\mathsf{PH}_{\text{same-model}}$
$\text{ACO}_{\text{undir}}$	$>$ ($\alpha = 1\%$)	$>$ ($\alpha = 1\%$)	$>$
$\mathsf{PH}_{\text{assign-pairs}}$		$>$ ($\alpha = 1\%$)	$<$ ($\alpha = 1\%$)
$\mathsf{PH}_{\text{assigns}}$			$<$ ($\alpha = 1\%$)

7.7. CSP

The CSP is a common problem in the car manufacturing industry [26]. The aim of the variant considered here is to place cars of different models in a production sequence such that the separation penalty between cars of the same model is minimised.[10] One way this can be formulated is as the allocation of sequence positions ($\mathfrak{C}_{\text{it}}$) to different models of cars ($\mathfrak{C}_{\text{res}}$), which accordingly lends itself to solutions with the same form as those to the GAP. The objective function of this formulation is

$$\text{Minimise} \sum_{i=1}^{M} \sum_{j=1}^{|D(i)|-1} \sum_{k=j+1}^{|D(i)|} P(|A(i,k) - A(i,j)|, i), \tag{2}$$

where $D(i) = \{j \in \mathfrak{C}_{\text{it}} \,|\, k \in [1, N],\ \mathfrak{s}[k] = i,\ j = I(k)\}$ is the set of sequence positions assigned to model i, with k being an integer, $\mathfrak{s}[k] \in \mathfrak{C}_{\text{res}} = \{1, \ldots, M\}$ the model assigned to sequence position $I(k)$, and $I(k)$ being the ith sequence position to be assigned during construction. $A(i, j) \in D(i)$ is the jth sequence position assigned to model i, $P(i, j)$ is the separation penalty for the jth model separated by i places in the sequence, N is the number of cars and M is the number of models.

In effect, this formulation is equivalent to that of a GAP in which each model's "capacity" is the number of cars of that model, while each sequence position requires one unit of its assigned model's "capacity". Consequently, it has no representation bias but does have a construction bias, so no pheromone model for it can be a CBS.

The CSP instances studied are those used by Smith et al. [26] and range in size from 20 to 80 cars, with cars in each instance divided unevenly into four models. Four pheromone models were studied with this problem, $\mathsf{PH}_{\text{assigns}}$, $\mathsf{PH}_{\text{suc-res}}$ and two higher order models. The first of these, $\mathsf{PH}_{\text{same-model}}$, has $C \subset \mathfrak{C}_{\text{it}} \times \mathfrak{C}_{\text{it}}$ and models two sequence positions being assigned the same model, which examination of the objective function reveals has some impact on solution cost. However, it also has the disadvantage that completely different solutions may be represented by the same subset of solution characteristics and so was predicted to perform badly [7]. The second higher order pheromone, $\mathsf{PH}_{\text{assign-pairs}}$, has $C \subset \mathfrak{C}_{\text{it}}^2 \times \mathfrak{C}_{\text{res}}$ and models pairs of assignments, where both sequence positions are assigned the same model. The potentially multiple pheromone values from $\mathsf{PH}_{\text{same-model}}$ and $\mathsf{PH}_{\text{assign-pairs}}$ are combined by taking the mean for the relevant solution characteristics. When no sequence positions have yet been assigned a candidate car model, and hence neither higher order model can be used, $\mathsf{PH}_{\text{assigns}}$ is used to influence the decision. Both $\mathsf{PH}_{\text{assigns}}$ and $\mathsf{PH}_{\text{assign-pairs}}$ are identity-oriented pheromones, although as $\mathsf{PH}_{\text{assign-pairs}}$ models most closely those solution characteristics that have a direct impact on cost (i.e., pairs of sequence positions assigned to the same model, combined with which model they are assigned) it was predicted to perform best [7].

As with the QAP and GAP, there exists a choice concerning the order in which sequence positions are assigned. Two alternative orders were considered: numerical order and a dynamic, randomised order. As $\mathfrak{C}_{\text{res}} \times \mathfrak{C}_{\text{res}}$ requires a static assignment order to be sensibly applied it is only involved in comparisons within results for the former.

Table 9 shows the results of Mann–Whitney tests comparing alternative approaches when using $\mathcal{MM}$AS with a dynamic, randomised assignment order, which was the best performing algorithm combination. Although not shown in the table, it should be noted that when assigning positions in numerical order under ACS, the performance of

[10] The CSP may also be modelled as a constraint satisfaction problem, where each station on a production line can support a limited number of cars of the same model in sequence, as in Gottlieb et al. [30].

$\mathsf{PH}_{\mathsf{assign-pairs}}$ was worse than *all* other alternatives, with further analysis revealing that the algorithm was unable to converge. The performance of all approaches was improved when using the randomised order. Using this order with ACS, $\mathsf{PH}_{\mathsf{assign-pairs}}$ outperforms $\mathsf{PH}_{\mathsf{assigns}}$ on large instances (60–80 cars), although the reverse is true under $\mathcal{MM}$AS.

When assigning positions in numerical order with the best performing ACO algorithm, $\mathcal{MM}$AS, the alternatives may be ranked in terms of solution cost as

$$\mathsf{PH}_{\mathsf{assigns}} \prec \mathsf{PH}_{\mathsf{same-model}} \prec \mathsf{PH}_{\mathsf{assign-pairs}} \prec \mathsf{PH}_{\mathsf{suc-res}} \prec \mathrm{ACO}_{\mathrm{undir}}$$

while when using the randomised order with $\mathcal{MM}$AS they may be ranked as

$$\mathsf{PH}_{\mathsf{assigns}} \prec \mathsf{PH}_{\mathsf{assign-pairs}} \prec \mathsf{PH}_{\mathsf{same-model}} \prec \mathrm{ACO}_{\mathrm{undir}}.$$

It should be noted that when using the randomised order, both $\mathsf{PH}_{\mathsf{assigns}}$ and $\mathsf{PH}_{\mathsf{assign-pairs}}$ often achieved minimum solution costs within 1–5% of the optimum, even on large instances, while the alternatives were significantly worse, with costs often greater than 20% more than the optimum.

Notably, the best performing pheromone for this problem was $\mathsf{PH}_{\mathsf{assigns}}$ rather than $\mathsf{PH}_{\mathsf{assign-pairs}}$, although both performed well. Although both are identity-oriented, it has been suggested previously that $\mathsf{PH}_{\mathsf{assign-pairs}}$ is more appropriate as it represents how the cost of a particular car model at a particular location in the production sequence is dependent on the locations of other cars of the same model. In contrast, $\mathsf{PH}_{\mathsf{assigns}}$ learns the utility of placing a particular model at a particular position in the production sequence regardless of the location of other cars of the same model. However, if ACO is considered to be searching for a single high quality solution, then modelling which car model belongs at each position in the sequence does largely model characteristics that determine solution cost. Examination of the distribution of solution costs produced by $\mathsf{PH}_{\mathsf{assigns}}$ reveals that, on average, almost 40% of solutions produced have the same cost, suggesting that it does converge to a single good solution for each instance to which it is applied.

7.8. Summary of results

Comparisons of alternative pheromones for the TSP, MKP, GSP, QAP, GAP and CSP show that, in most cases, identity-oriented pheromone models outperform representation-oriented alternatives. In the TSP, GSP and GAP, these results hold even if a heuristic bias or, in the case of the TSP, local search is used. Interestingly, results for the MKP show that when a heuristic bias is not used, or when the ACO algorithm is $\mathcal{MM}$AS, a $\mathsf{PH}_{\mathsf{pos}}$ model can outperform the intuitively more appropriate $\mathsf{PH}_{\mathsf{items}}$ model. Indeed, $\mathsf{PH}_{\mathsf{pos}}$ has never been proposed for use with the MKP. However, when using ACS with a heuristic bias, both $\mathsf{PH}_{\mathsf{pos}}$ and its second order counterpart outperform $\mathsf{PH}_{\mathsf{pos}}$.

Results for the QAP reveal that when local search is not used, the suggested pheromone can outperform an alternative pheromone as well as an undirected search. When local search is used, all three alternatives perform equivalently on smaller instances, finding the optimum in every case, although the suggested pheromone performs better on two of the three largest instances with which local search was used. When combining either pheromone model for the QAP with local search, very fast convergence to one solution was observed, especially when using the pheromone associated with assignments. This suggests that when local search is used a number of short runs of ACO may be a more effective way of sampling the search space and may lead to better performance than an equivalent number of local searches starting from randomly constructed solutions.

With the CSP, the two identity-oriented pheromones outperform the two alternatives when a dynamic, randomised assignment order is used. The suggested pheromone, $\mathsf{PH}_{\mathsf{assign-pairs}}$, performs very badly when positions are assigned in numerical order, while the first order pheromone associated with assignments still performs well. When a randomised assignment order is used, $\mathsf{PH}_{\mathsf{assign-pairs}}$ in most cases performs slightly worse than its first order counterpart, suggesting that the higher level information it provides may not be of benefit to the algorithm. Thus, even though the first order pheromone does not learn exactly those solution characteristics that contribute to cost (as an assignment on its own makes no direct contribution to cost) it still appears to offer the most effective way to learn which assignments to make to produce a good solution.

Based on these results, Table 10 shows a ranking of the alternative pheromone models examined in this study when the best performance was observed. Hence, although $\mathsf{PH}_{\mathsf{items}}$ can outperform $\mathsf{PH}_{\mathsf{pos}}$ under certain conditions, for instance, $\mathsf{PH}_{\mathsf{pos}}$ is ranked more highly as when both models performed best it was the better of the two.

Table 10
Summary of pheromone model comparative performances

Problem	Pheromone model/algorithm in rank order				
TSP	PH_{suc}*	PH_{pos}	ACO_{undir}		
MKP	PH_{pos}	PH_{items}**	PH_{pairs}*	PH_{suc}	ACO_{undir}
JSP, GSP	PH_{rel}*	PH_{pos}	ACO_{undir}	PH_{suc}	
OSP	PH_{rel}*	PH_{pos}	PH_{suc}	ACO_{undir}	
QAP	$\mathsf{PH}_{assigns}$*	$\mathsf{PH}_{suc-res}$	ACO_{undir}		
GAP	$\mathsf{PH}_{assigns}$*	$\mathsf{PH}_{suc-res}$	ACO_{undir}		
CSP	$\mathsf{PH}_{assigns}$*	$\mathsf{PH}_{assign-pairs}$**	$\mathsf{PH}_{same-model}$	$\mathsf{PH}_{suc-res}$	ACO_{undir}

Models that are identity-oriented are marked by an asterisk (∗). If there is more than one such model used with a problem, the model suggested by Montgomery et al. [10] is marked by a double asterisk (∗∗). Models are ranked left to right from best to worst.

8. Conclusions

The ACO metaheuristic can produce good quality solutions to CO problems if it is applied correctly. In particular, the pheromone model used must be chosen carefully. The nature of the solution construction process is such that, due to problem constraints, unavoidable biases exist in constructive algorithms for some problems. These in turn lead to biases in the pheromone models that may be used in ACO algorithms for those problems. The possible presence of these biases must be considered by ACO algorithm developers when choosing how solutions are built and which pheromone model to use.

At the lowest level, constructed solution biases will determine the baseline probability of a particular solution characteristic being updated. However, given the influence of these low level biases appears negligible on larger instances, the relative number of sequences each solution characteristic describes may be the ultimate determinant of how often it is reinforced (at least during the early stages of an ACO algorithm when decisions are made more randomly). This latter issue is addressed by Blum's [13] definition of a CBS, which is a problem–pheromone combination in which every solution characteristic appears the same number of times. If a pheromone model applied to a problem is not a CBS, that pheromone will exhibit a bias. It has been shown that the presence of an underlying construction bias will prevent any pheromone model from being a CBS. Additionally, when a representation bias is present, a pheromone may be a CBS yet not be free from solution bias.

In certain problems, notably the JSP, the problem's structure may serve to bias a search towards certain kinds of solution. When the solution characteristics modelled by a pheromone model are associated strongly with one kind of solution or another, that pheromone model will consequently show a bias towards those kinds of solutions.

Unlike the JSP, the MKP and GAP do not appear to have a clearly exploitable structure. In the MKP, solution characteristics associated with the inclusion of items with few resource requirements will be updated most frequently. However, even though items with large resource requirements are often more valuable and so would be expected to be reinforced less often, the majority of solution characteristics appear in a wide range of solutions, and so will not necessarily bias a search towards good or bad solutions. Nevertheless, the addition of a heuristic bias that favours high value and consequently infrequent solution characteristics improves the algorithm's performance. In the GAP, solution characteristics associated with assignments that utilise less of an agent's capacity will be updated most frequently, but in the absence of a strong relationship between capacity utilisation and value most solution characteristics are not strongly associated with either good or bad solutions. The observed behaviour of an ACO algorithm for this problem suggests that the algorithm does converge to solutions containing characteristics that are more frequent, although given the tight constraints of this problem such characteristics are also those most likely to lead to a feasible solution.

If the nature of the biases in a collection of alternative pheromone models is known then this knowledge may be used to select the model with the most advantageous bias. However, as the examples examined in this paper illustrate, often no predictable pattern can be identified, in which case identity-oriented pheromone models may offer the safest alternatives [10].

The relative performance of alternative pheromone models for six well-known problems was compared in order to test the hypothesis that identity-oriented pheromones perform best, particular if the characteristics they model are strongly related to solution cost or value. For all problems except the MKP the best performing models were identity-oriented, with results for the MKP requiring further investigation as the best performing pheromone model was one that has never been suggested as a valid alternative and which intuitively appears inappropriate. Results for the CSP suggest that modelling solution characteristics that are directly related to solution cost may not always be best. Possibly in this case it is because doing so creates a more complex higher order model that consequently learns more slowly than its first order counterpart. The utility of higher order models is discussed by Montgomery [31].

Finally, it should be noted that the bias analysis techniques suggested in this paper represent an initial step towards the predicted reaction of a pheromone model to the underlying biases in a problem. The development of effective techniques in this area is non-trivial, but is critically important to the successful future application of ACO in new problem domains.

References

[1] Zlochin M, Dorigo M. Model-based search for combinatorial optimization: a comparative study. In: Proceedings of the seventh international conference on parallel problem solving from nature (PPSN2002). 2002. p. 651–62.
[2] Blum C, Sampels M. Ant colony optimization for FOP shop scheduling: a case study on different pheromone representations. In: Proceedings of the 2002 congress on evolutionary computation. 2002. p. 1558–63.
[3] Dorigo M, Stützle T. The ant colony optimisation metaheuristic: algorithms, applications and advances. In: Glover F, Kochenberger G, editors. Handbook of metaheuristics. Boston, MA: Kluwer Academic Publishers; 2002. p. 251–85.
[4] Blum C, Dorigo M. Search bias in ant colony optimization: on the role of competition-balanced systems. IEEE Transactions on Evolutionary Computation 2005;9(2):159–74.
[5] Blum C, Sampels M. When model bias is stronger than selection pressure. In: Merelo-Guervós JJ, et al., editors. Proceedings of the seventh international conference on parallel problem solving from nature (PPSN2002). Granada, Spain; 2002. p. 893–902.
[6] Merkle D, Middendorf M. Modelling the dynamics of ant colony optimization algorithms. Evolutionary Computation 2002;10(3):235–62.
[7] Montgomery EJ. Solution biases and pheromone representation selection in ant colony optimisation. Ph.D. thesis, Bond University, 2005.
[8] Montgomery J, Randall M, Hendtlass T. Search bias in constructive metaheuristics and implications for ant colony optimisation. In: Dorigo M, et al., editors. Proceedings of the fourth international workshop on ant colony optimization and swarm intelligence, ANTS 2004, Brussels, Belgium; 2004. p. 390–7.
[9] Montgomery J, Randall M, Hendtlass T. Structural advantages for ant colony optimisation inherent in permutation scheduling problems. In: Ali M, Esposito F, editors. Proceedings of the 18th international conference on industrial and engineering applications of artificial intelligence and expert systems (IEA/AIE 2005). Bari, Italy; 2005. p. 218–28.
[10] Montgomery J, Randall M, Hendtlass T. Automated selection of appropriate pheromone representations in ant colony optimisation. Artificial Life 2005;11(3):269–91.
[11] Stützle T, Dorigo M. ACO algorithms for the traveling salesman problem. In: Miettinen K et al., editor. Evolutionary algorithms in engineering and computer science. New York: Wiley; 1999. p. 163–83.
[12] Blum C, Dorigo M. The hyper-cube framework for ant colony optimization. IEEE Transactions on Systems, Man and Cybernetics, Part B 2003;34(2):1161–72.
[13] Blum C. Theoretical and practical aspects of ant colony optimization. Dissertations in artificial intelligence, vol. 282. Nieuwe Hemweg, The Netherlands: IOS Press; 2004.
[14] Birattari M, Di Caro G, Dorigo M. Toward the formal foundation of ant programming. In: Dorigo M, Di Caro G, Sampels M, editors. Proceedings of the third international workshop on ant algorithms, ANTS 2002, Brussels, Belgium; 2002. p. 188–201.
[15] Maniezzo V. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. INFORMS Journal on Computing 1999;11(4):358–69.
[16] Maniezzo V, Milandri M. An ant-based framework for very strongly constrained problems. In: Dorigo M, Di Caro G, Sampels M, editors. Proceedings of the third international workshop on ant algorithms, ANTS2002, Brussels, Belgium; 2002. p. 222–7.
[17] Colorni A, Dorigo M, Maniezzo V, Trubian M. Ant system for job-shop scheduling. JORBEL 1994;34(1):39–53.
[18] Blum C, Sampels M. An ant colony optimization algorithm for shop scheduling problems. Journal of Mathematical Modelling and Algorithms 2004;3(3):285–308.
[19] Beasley JE. OR-Library, ⟨http://people.brunel.ac.uk/~mastjjb/jeb/info.html⟩, 2005.
[20] Roli A, Blum C, Dorigo M. ACO for maximal constraint satisfaction problems. In: Proceedings of the fourth metaheuristics international conference. Porto, Portugal; 2001. p. 187–92.
[21] Socha K, Knowles J, Sampels M. A $\mathcal{MAX}$–$\mathcal{MIN}$ ant system for the university course timetabling problem. In: Dorigo M, Di Caro G, Sampels M, editors. Proceedings of the third international workshop on ant algorithms, ANTS 2002, Brussels, Belgium; 2002. p. 1–13.
[22] Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1997;1(1):53–66.
[23] Stützle T, Hoos H. The $\mathcal{MAX}$–$\mathcal{MIN}$ ant system and local search for combinatorial optimization problems: towards adaptive tools for combinatorial global optimization. In: Voss S et al., editor. Meta-heuristics: advances and trends in local search paradigms for optimization. Boston, MA: Kluwer Academic Publishers; 1998. p. 313–29.
[24] Bertsekas DP. Nonlinear programming. 2nd ed, Belmont, MA: Athena Scientific; 1999.
[25] Meuleau N, Dorigo M. Ant colony optimization and stochastic gradient descent. Artificial Life 2002;8(2):103–21.
[26] Smith K, Palaniswami M, Krishnamoorthy M. A hybrid neural network approach to combinatorial optimisation. Computers and Operations Research 1996;73:501–8.

[27] Reinelt G. TSPlib—a traveling salesman problem library, ⟨ http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/⟩, 2004.
[28] Burkard RE, Çela E, Karisch SE, Rendl F. QAPLIB—a quadratic assignment problem library, ⟨ http://www.seas.upenn.edu/qaplib/⟩, 2005.
[29] Randall M. Heuristics for ant colony optimisation using the generalised assignment problem. In: Proceedings of the 2004 congress on evolutionary computing. Portland, OR, USA; 2004. p. 1916–23.
[30] Gottlieb J, Puchta M, Solnon C. A study of greedy, local search, and ant colony optimization approaches to car sequencing problems. In: Raidl G, et al., editors. Proceedings of the evoworkshops 2003. Essex, UK; 2003. p. 246–57.
[31] Montgomery J. Higher order pheromone models in ant colony optimisation. In: Dorigo M, et al., editors. Proceedings of the fifth international workshop on ant colony optimization and swarm intelligence, ANTS 2006, Brussels, Belgium; 2006. p. 428–35.