Automated Selection of Appropriate Pheromone Representations in Ant Colony Optimization

James Montgomery
Marcus Randall
Faculty of Information Technology
Bond University
QLD 4229
Australia
jmontgom@bond.edu.au
mrandall@bond.edu.au

Tim Hendtlass
Centre for Intelligent Systems and Complex Processes
Swinburne
University of Technology
VIC 3122
Australia
thendtlass@swin.edu.au

Abstract Ant colony optimization (ACO) is a constructive metaheuristic that uses an analogue of ant trail pheromones to learn about good features of solutions. Critically, the *pheromone representation* for a particular problem is usually chosen intuitively rather than by following any systematic process. In some representations, distinct solutions appear multiple times, increasing the effective size of the search space and potentially misleading ants as to the true learned value of those solutions. In this article, we present a novel system for automatically generating appropriate pheromone representations, based on the characteristics of the problem model that ensures unique pheromone representation of solutions. This is the first stage in the development of a generalized ACO system that could be applied to a wide range of problems with little or no modification. However, the system we propose may be used in the development of *any* problem-specific ACO algorithm.

Keywords
Ant colony optimization, pheromone, metaheuristic, combinatorial optimization

1 Introduction

For many years the operations research community has sought an efficient integer solver capable of solving a range of combinatorial optimization problems (COPs) with little or no modification [1]. Traditional operations research techniques such as branch and bound have exponential worst case running times, prompting the development of a small number of generalized metaheuristics, as detailed in [38]. Recent years have also seen the emergence of *hyperheuristics* [10], metaheuristics that are used to select, in a problem-independent manner, appropriate problem-specific heuristics to use at each point during search. This article describes part of a larger effort to produce a generalized *constructive* metaheuristic [33, 36], based on the increasingly popular ant colony optimization (ACO) approach (see [15] and [17]). The work described herein is not, strictly speaking, part of a hyperheuristic, as the approach proposed incorporates broad knowledge of a number of domains to make decisions about how to adapt ACO. In effect, it concerns the development of a self-adapting ACO algorithm rather than a higher level search of competing ACO algorithms.

The ACO approach belongs to the class of model-based search (MBS) algorithms [44]. In a MBS algorithm, new solutions are generated using a parameterized 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 used in ACO is known as a *pheromone representation*, and is an artificial analogue of the chemical used by real ants to mark trails from the nest to food sources. In ACO, pheromone information is typically associated with the solution components used by artificial ants to construct new solutions, guiding their decisions. The pheromone representation is one of the most important design choices when adapting ACO to a given problem [7, 17]. Typically, it is chosen in an ad hoc way, based on what *appears* to best suit a given problem. Intuitive choices often work quite well in practice [17]. However, in a number of cases pheromones have been used that represent solutions multiple times, increasing the apparent size of the search space and potentially misleading ants as to the true learned value of solutions. For instance, the most common pheromone representation used with shop scheduling problems associates pheromone with the absolute position of scheduled operations in the solution vector, which in most instances allows the same solutions to be represented by different sets of pheromone values. This pheromone is less effective than an alternative pheromone that does not represent solutions multiple times [7]. A more rigorous and systematic approach to selecting pheromones may be adopted that ensures unique representation of solutions, improving the consistency with which ACO is applied and potentially leading to better results. In this article, we describe a novel system for producing appropriate pheromone representations based on the characteristics of the combinatorial problem to solve. This system may be applied in a generalized solver as well as in the development of problem-specific ACO algorithms.

In Section 1.1 we review the small number of studies that have compared alternative pheromone representations. Section 2 contrasts the formal description of ACO with its application in practice. Next, in Section 3, we introduce a formal notation for describing pheromone representations. Sections 4 and 5 explore the requirement of unique solution representation and the role of parsimony in the pheromone representation. Section 6 describes our approach to deriving appropriate pheromone representations for different problems. Section 7 discusses important areas of future investigation.

1.1 Comparative Pheromone Studies

There have been a small number of comparative studies on alternative pheromone representations. In the main, these have used the observed performance of alternative intuitive pheromone choices to infer the most appropriate pheromone representation. Hence, their results are restricted to those problems studied and some closely related problems.

Blum [4] studies two pheromone representations for the edge-weighted k-cardinality tree problem in which a tree of k edges of minimum weight from some graph is sought. One representation associates pheromone with the edges in the tree, while the other associates pheromone with pairs of edges. Given the same amount of execution time, the latter produces fewer solutions due to its increased computational overhead, leading to the conclusion that the former is a better choice for this problem.

Roli, Blum, and Dorigo [39] describe a maximal constraint satisfaction ACO algorithm and compare three pheromone representations that associate pheromone with the assignment of values to variables, pairs of variable-value bindings adjacent in the solution, and all pairs of variable-value bindings in a solution, respectively. The first and last performed similarly, but, due to the increased computational overhead for the last, the first representation is promoted as the best suited to this problem.

Socha, Knowles, and Sampels [41] consider two alternative pheromone representations for a university course timetabling problem with the aim of minimizing a number of soft constraints. The first representation associates pheromone with the assignment of an event to a time, while the second considers pairs of events assigned to the same time. While the latter was considered to be more appropriate for timetabling, the former was found to be more effective on this particular problem. Possibly this is because this is not a classic timetabling problem in which clashes must be minimized (only feasible timetables were constructed), but one where the actual times assigned to particular events have an important effect on the solution cost (by affecting the soft constraints).

Blum and Sampels [7] compared four pheromone representations for a generalized scheduling problem: three from the literature, and a novel representation chosen to closely match the schedules represented rather than the structure of ants' solutions. Their novel pheromone representation performed best. Indeed, in a later study, they found that one of the representations from the literature introduces an unfavorable bias into the search [8].

The first two studies suggest that too much information in a pheromone representation reduces its computational efficiency. The last two studies suggest that choosing a pheromone that models some aspect of the problem that has a strong influence on solution quality may be the most effective. Section 4 presents some of the reasons underpinning these results. Knowledge of these underlying causes aids in the development of a systematic approach to the selection of pheromone representations. This systematic approach may then be used with problems to which ACO has not previously been applied.

2 ACO, Construction Graphs, and the Pheromone

The earliest ACO algorithm, Ant System (AS) [16], was applied first to the traveling salesman problem (TSP), which has a strong similarity in structure to the real world shortest path problem faced by ants when foraging for food. A number of improved ant algorithms were inspired by AS, and in the late 1990s the common aspects of these were combined into the formal description of the ACO metaheuristic (see, e.g., [15]). ACO algorithms (instances of the ACO metaheuristic) are intended to find minimum cost paths over a graph $G = (C, L, W)$ while respecting a set of constraints Ω [14, 15]. In this formulation $C = (c_1, c_2, \ldots, c_n)$ is a finite set of problem *components*, $L = (l_{c_i c_j} \mid c_i, c_j \in C)$ is a finite set of possible *connections* between the elements of C, and W is a set of weights associated with the components C or the connections L or both. Hence, for the TSP C is the set of cities, L is the set of links between cities, and W is the weighting of the links. This definition admits pheromone representations that do not associate pheromones with the edges of the graph. We broaden the interpretation of solution components in Section 3.

A recent effort to describe a generalized version of AS is the graph-based ant system (GBAS) [23]. In GBAS, ants create walks over a *construction graph* that represents an encoded form of solutions to a particular problem. The pheromone is associated with the edges of this graph. In essence, the construction graph is an abstract form of the solution, encoding the solution construction process rather than directly representing solutions. Walks in this construction graph are mapped onto feasible solutions via a problem-specific function Φ. The GBAS is as yet only a theoretical system, and no attempt has been made to describe how the function Φ may be input to the system, a critical step in implementation.

Di Caro and Dorigo's [13] AntNet for dynamic routing in telecommunication networks is somewhat generalized in that it may be applied to any packet-switched network. As networks clearly display a graph-based structure, it also associates pheromone with the links between nodes in the network.

Birattari, Di Caro, and Dorigo's [3] *ant programming* is another notable effort to describe ant algorithms in a generalized fashion. Ant programming combines ideas from ACO and dynamic programming and thus emphasizes the role of the state graph (i.e., the decision tree) that describes the chain of constructive decisions leading to each partial solution. In essence, ant programming is an approach to describing any potential ant algorithm, including existing ACO algorithms, rather than a specific approach to a generalized system. Although the role of the state graph is a central part of ant programming, Birattari et al. suggest that a pheromone representation should actually be tailored to suit the particular solutions represented, rather than the states and transitions in the state graph.

As ACO is based on the foraging behavior of ants—clearly a shortest path problem—it may seem natural and even essential that the same restriction be imposed on applications of ACO to various optimization problems. Certainly, the development of an appropriate graph representation is considered to be of crucial importance in much of the ACO literature (e.g., [2, 11, 14–17, 20, 22, 39, 41, 43]). However, ACO algorithms have increasingly been applied to problems that show a marked divergence from classic shortest path problems like the TSP.

Consequently, the graph representation of these problems offers little or no assistance in choosing an appropriate pheromone representation. This has led to the ad hoc application of several novel pheromone representations.

A typical approach to developing a graph representation for a COP involves identification of the components from which solutions are constructed and then determining how those components should be connected. This can be difficult for many COPs and often has lead to the development of many ACO applications that diverge from the traditional construction graph approach.

The multiple knapsack (MKP), set covering (SCP), and set partitioning (SPP) problems, collectively called *subset problems*, have the simplest solution structure of any problem type, consisting of a set of items. A graph representation for these problems might use C as the set of items, with L fully connecting the elements of C. Each node visited in a walk in this graph would be included in the solution. However, as Leguizamón and Michalewicz [26] suggest, there is no real concept of a path in these problems. Hence, even if pheromone is associated with the items (an intuitive choice used by Leguizamón and Michalewicz), considering these problems in a terms of a graph is quite artificial.

For problems involving assignment of items to groups or resources, such as the quadratic assignment problem (QAP), generalized assignment problem (GAP), and frequency assignment problem (FAP) [28], two distinct types of component may be identified, items and resources. For instance, in the GAP, items represent various projects while resources represent the agents that may work on projects. In a review of ACO algorithms for the QAP, Stützle and Dorigo [43] suggest that a suitable graph representation for the QAP has C as the set of locations and facilities, with L fully connecting these components.[1] The graph representation of such problems would in fact be bipartite, as facility-facility and location-location links are meaningless for such problems. Moreover, given a walk in this graph corresponding to a feasible solution, only half the edges represent assignments, while the other half are superfluous. Indeed, ACO algorithms for the QAP do not use this graph representation [43], instead representing solutions as permutations of either facilities or locations, where position indicates the location (or facility) to which they are assigned, and pheromone is associated with the assignments. ACO algorithms for other assignment problems such as the FAP and GAP use pheromone representations that also suggest an alternative solution representation to the graph described above [27, 28].

Costa and Hertz [12] describe an ACO algorithm for the graph coloring problem (GRAPH), a special kind of assignment problem, in which pheromone is associated with pairs of non-adjacent nodes assigned the same color. The same pheromone is used for the bin packing (BIN) and cutting stock (CSP) problems [18]. While this pheromone representation is intuitively appropriate for these problems, it cannot be easily reconciled with walks in any graph representation of the problem.

Bauer, Bullnheimer, Hartl, and Strauss [2] describe an ACO algorithm for the single machine total tardiness problem (SMTTP), which considers two alternative graph representations. The first, a GBAS-inspired state-oriented construction graph, was not used on account of its enormous size, that is, $O(2^n)$ nodes when scheduling n operations, prompting the development of a greatly simplified construction graph. Pheromone is associated with the absolute position of nodes in walks in this simplified graph.

Socha, Knowles, and Sampels' [41] ACO algorithm for a university course timetabling problem represents solutions as walks in a construction graph where each node represents the assignment of an event (e.g., a class) to a time slot. The simpler of the two pheromone representations they consider (see Section 1.1) associates pheromone with the nodes of this graph and so is equivalent to the pheromone representation used on assignment problems. The more complex pheromone representation is highly similar to the one used for graph coloring. Neither of these requires nor reflects the graph representation of the problem.

Despite departing from the original construction graph approach in both the problem and the pheromone representation used, the ad hoc application of ACO has often been highly effective [17]. Indeed, ACO is only an analogue of ant behavior, and non-graph-based models can still allow the

1 This approach may also be applied to other assignment problems.

desired stigmergic behavior to emerge. However, the ad hoc approach has produced some combinations of COPs and pheromone representations with inherent flaws that may impair ACO's performance. Investigation of these problems can lead to a more systematic approach to the selection of appropriate pheromone representations. The next section formalizes the description of pheromone representations as background to the remainder of the article.

3 Pheromone Representations

An important foundation for the automated selection of pheromone representations is an appropriate way of describing those representations, which we introduce here.

To guide ants as they construct solutions, a pheromone representation must map certain identifiable features of solutions to pheromone values. Hence, the nature of a pheromone representation is largely determined by the features it describes. In the following, $C \subset \mathbb{Z}$ is the set of entities that may be combined to form a solution, called *components* by Dorigo, Bonabeau, and Theraulaz [14]. This differs from more recent interpretations of the term *solution component* (see, e.g., [6]) that consider it to be a decision variable plus its value. For a given combinatorial optimization problem, often several alternative sets of decision variables may be defined; the decision variables referred to here are those that most closely reflect the way ants construct solutions to the problem in question. For example, in the TSP a variable x_i, representing the city placed after city i, may take the value $j \in C \setminus \{i\}$, thereby representing the presence of edge (i, j) in the solution. This is the interpretation of the solution component used here.

Using this definition of solution component, Blum and Dorigo [6] proceed to describe ACO algorithms in terms of fully connected construction graphs, where each node corresponds to a solution component.[2] This definition requires that pheromone be associated with the nodes in such a construction graph, that is, the solution components. However, a distinction must be made between the *component(s)* added at each constructive step in an ACO algorithm and the nature of the decision variable(s) involved and the pheromone representation required to model the resultant binding of a value to a decision variable.

Some problems involve a number of different entities (such as facilities and locations in the QAP), in which case C consists of a number of subsets, $C_1, \ldots, C_n$, representing each of these entities, and which form a cover of C. A solution component can be constructed by combining elements of these subsets. For instance, given a set of cities C, $C \times C$ could represent the set of edges in the TSP, while given a set of facilities C_1 and a set of locations C_2, $C_1 \times C_2$ could represent the set of facility-location assignments in the QAP. Where a pheromone representation makes use of the absolute position of elements from C in a solution vector, the set of solution positions is denoted by P.

We further expand the notion of a solution component to the more general concept of a *solution characteristic*. An individual characteristic may correspond to the presence of a particular solution component, or to some broader feature of solutions produced by a number of solution components. For instance, in the TSP, $(i, j) \in C \times C$ is a solution component that is identical to the solution characteristic represented by the pheromone used for the TSP. An example of the latter kind of solution characteristic is found in a pheromone representation used for the graph coloring problem [12], where the solution characteristics modeled are pairs of nodes being assigned the same color, while the solution components are node-color assignments. Hence, the definition of a solution characteristic may correspond to an alternative set of decision variables for the problem being solved. In the case of the graph coloring problem, solutions are produced by the assignment of colors to nodes, which suggests that decision variables have the form x_i, representing the color assigned to node i. Yet a pheromone representation that models pairs of nodes being assigned the same color defines decision variables with the form x_{ij}, which is 1 if nodes i and j are assigned the

2 When alternative bindings of the same decision variable to different values appear in such a construction graph, the graph cannot be fully connected, as that would imply that an ant may select more than one value for a decision variable.

same color, and 0 otherwise. These two alternative definitions of the decision variables are not contradictory. Indeed, pheromone information associated with the latter decision variables is used to decide the value of the former kind, and represents a higher order representation of solutions to the problem.

Where solution characteristics correspond to solution components, a pheromone representation is said to be *first order* [7]. In such cases, a single pheromone value from the representation is used to influence an ant's decision regarding a single solution component. Higher order representations involve subsets of solution components. For instance, a second order pheromone representation often indicates the utility of having a pair of solution components in the same solution. Higher order pheromone representations may emerge in two ways. Given an existing first order pheromone M, the nth order pheromone representation may be obtained by transforming it into M^n. Alternatively, when the solution characteristics we wish to model relate many parts of the solution to each other, a higher order pheromone forms naturally as a consequence of having to combine information from each of these relationships. In the latter case, unlike the former, it is often impossible to use the underlying first order representation independently. In both cases, the resulting representation is denoted by $X \times M; M^n$, where X represents the set of all partial solutions. The first part describes the *observed* pheromone representation, where the pheromone associated with adding a solution characteristic from M is contingent on a partial solution from the set X. The observed pheromone representation provides a single pheromone value for each candidate solution component and hence is equivalent to a first order representation. Therefore, ants make decisions based on the values in this observed pheromone representation. The second part describes the underlying pheromone representation M^n, from which pheromone values are aggregated[3] to produce the observed pheromone representation. An example appears below. Strictly speaking, the nth order pheromone representation generated from M includes only higher order solution characteristics (and hence pheromone values) for n-tuples consisting of *n distinct* elements from M, although this is denoted by M^n.[4] The structure of a typical nth order pheromone is described formally in Definition 1.

Definition 1: *The nth order pheromone representation derived from the first order pheromone representation M is* $\{(i_1, \ldots, i_n) \mid i_1, \ldots, i_n \in M, \quad i_1 \prec \cdots \prec i_n\} \subset M^n$, *where* $i_j \in M$ *is a solution component from M, and* $\prec$ *is an arbitrary but fixed order imposed on the elements of M. This is denoted by* $X \times M; M^n$.

An example of a naturally occurring second order pheromone is that used for the graph coloring problem [12]. For this problem, a $X \times C_1 \times C_2; (C_1)^2$ pheromone is used, where $(C_1)^2$ represents a pair of nodes being assigned the same color. Hence, the pheromone representation naturally relates a number of solution components. Although Definition 1 cannot be applied directly to this second order pheromone, the standard M–M^2 relationship can still be seen by considering a typical second order pheromone where solution characteristics are pairs of node-color assignments, $X \times C_1 \times C_2; (C_1 \times C_2)^2$, that is, where solution characteristics correspond to the solution components used to build solutions. Given that the actual colors assigned to nodes do not affect the cost of solutions—only what nodes are in the same color group—all references to actual colors in the underlying $(C^1 \times C^2)^2$ pheromone may be removed, producing the simpler $(C_1)^2$ pheromone. We refer to this second order pheromone as a *grouping pheromone*.

Define the set of allowable pheromone values by $T \subset \mathrm{R}^+$, with individual pheromone values denoted by $\tau(i)$, where i is the solution characteristic being modeled. We abbreviate $M \mapsto T$ to M. Table 1 summarizes a number of commonly used pheromone representations.

Where the solution characteristic described by M is potentially ambiguous, it may be necessary to specify its proper interpretation. Consider representation 3 from Table 1, $X \times C; C^2$, where $(i, j) \in C^2$ can represent i and j being copresent in a solution or that i appears before j in the solution. These

3 Various methods may be used, such as taking the mean or minimum value.

4 This corresponds to the use of matrices to implement such representations, because of their fast access properties not possessed by sparse representations.

Table 1. Common pheromone representations. Example problems represent a sample of how representations have been used in the literature. In some cases different pheromones have been used with the same problem.

No.	Pheromone	Pheromone associated with:	Example problems
1	C	items present in solution	Knapsack, set covering
2	$C \times C$	one item succeeding another	TSP, shop scheduling
3a	$X \times C; C^2$	pairs of items present in solution	Maximum clique
3b		collection of items preceding another	Shop scheduling
4	$X \times C_1 \times C_2; (C_1)^2$	pairs of items in same/ different group	Graph coloring
5	$C \times P$	position of item in solution	Shop scheduling
6	$C_1 \times C_2$	assignment of resource in C_2 to item in C_1	Generalized and quadratic assignment

can be specified more clearly by denoting the representations as $X \times C; C^2$ (copresent) and $X \times C; C^2$ (preceding), respectively.

The notation introduced in this section is best suited to describing pheromone representations associated with solutions which are *collections* of solution components. That is, the pheromone representation may support a number of constructive algorithms for the problem in question, not just those that build solutions from a sequence of solution components. For instance, solutions to the TSP may be built as collections of links while still using a $C \times C$ pheromone.[5] The notation is thus suited to the majority of pheromone representations used in practice. However, in some ACO algorithms the pheromone representation and constructive process used are inextricably connected, and in such situations our notation is not easily applied. For instance, the shortest common supersequence problem (SCSP) [32] consists of creating a minimum length string of characters from some alphabet Σ (e.g., representing genes on a chromosome or machines in a production line) in such a way that it is a supersequence of a set L of other strings (i.e., any of the other strings may be produced by deleting characters from the solution). An ACO algorithm for this problem developed by Michel and Middendorf [32] constructs solutions in the following way. Throughout the solution construction process, the algorithm keeps track of how many characters from the start of each string in L have been included in the partial supersequence. At each step, the set of candidate characters consists of the next character to include from each string. A pheromone value is associated with each character in each string, suggesting a pheromone representation of $L \times I$, where I is the set of indices of characters within the strings of L. However, the decision to include a candidate character $c \in C$, where C is the set of characters, is based on a single pheromone value derived by summing pheromone values for each $(l, i) \in L \times I$, where i is the position of the next available character from the string l, such that the ith character of l is c. The pheromone representation is therefore most closely denoted by $X \times C; L \times I$. This differs from more typical pheromone representations in that

5 Clearly, building solutions to the TSP in this manner may not be desirable, as feasible solutions are no longer guaranteed; but it is certainly possible.

individual pheromone values have no meaning outside of the constructive process used, as each will contribute to the inclusion of a character at some point in a single solution's construction. In more typical pheromone representations the pheromone value associated with each solution component may be considered without that solution component being added to the solution.

3.1 Representation-Oriented and Identity-Oriented Pheromones

Broadly two approaches may be taken to derive a pheromone representation from the problem model used by ants to build solutions: representation-oriented or identity-oriented. The former produces pheromones that reflect some aspect of how solutions are represented (i.e., the arrangement of solution components), while the latter results in pheromones that describe what solutions are represented. Knowing the arrangement of solution components may identify solutions in some problems, but in many cases only indirectly indicates the identity of the solutions represented. For example, representing solutions to a subset problem as a linear list of items, a $C \times P$ pheromone may be used to learn where to place items in the structure. Hence, a decision variable x_i implied by this pheromone represents the item placed at location i. This indirectly indicates which items should be chosen. Using a C pheromone for a subset problem represents the identity of solutions quite separately from how they are represented and built by ants. The decision variables implied by this pheromone indicate whether an item is in the solution or not.

The two kinds of pheromone are not mutually exclusive. When a given solution structure represents each solution only once, any pheromone used with or derived from that structure will reflect both solution representation and identity. For instance, the $C \times C$ pheromone for the TSP represents both how solutions are represented (i.e., as permutations) and the identity of solutions as sets of edges.

A number of pheromone representations described in the literature are derived from structural aspects of solutions (see, e.g., [2, 11, 31, 42]). For many problems, these pheromones only indirectly identify solutions, potentially leading to poorer performance of the ACO algorithms that employ them. This is discussed in the next section.

4 Unique Solution Representation in Pheromones

When applying ACO to the TSP, an intuitive pheromone representation is $C \times C$, where C is the set of cities. This choice is seemingly a good one, as it associates pheromone with the solution feature that most directly contributes to cost (i.e., links), and indeed, like many other intuitive pheromone choices, it works quite well in practice [17]. Furthermore, this pheromone representation has an important feature. Each distinct solution to the TSP is represented by exactly one set of links and so, in the pheromone representation, by exactly one set of solution characteristics. (This is not the case, for instance, when using a $C \times C$ pheromone for a subset problem, as illustrated in Figure 1.) Hereafter, we say that pheromone representations with this feature have the property of unique representation, expressed by Definition 2.

Definition 2: *A pheromone representation M has the property of unique representation if each solution is represented by exactly one set of solution characteristics taken from M.*

There are two criteria for possessing this property. First, each distinct solution must be represented at least once, as any excluded solution could be the optimal one. As pheromone representations are derived from characteristics of the solution structures used, which will in any properly designed algorithm allow for any solution to be represented, this criterion is generally trivially met. In some pheromones, distinct solutions may share the same representation, as in one ACO algorithm for the static aircraft landing problem [37]. While such pheromones do not exclude any solutions, the sharing of a single representation by multiple solutions may be undesirable for problems where the same pheromone values are shared between solutions of quite

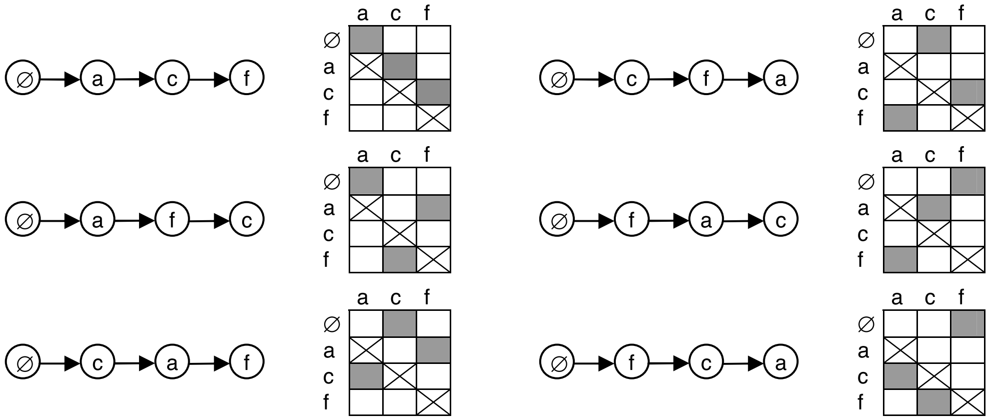

Figure 1. A small subset problem (in which the cost is associated with items) using the $C \times C$ pheromone representation. All possible ant solutions are shown for the subset $\{a, c, f\}$ taken from some larger set. For simplicity, the pheromone matrices shown only have entries for those items in this subset. ∅ is the artificial start point from which all ants begin. Shaded cells indicate the solution characteristics (pheromone values) corresponding to the adjacent solution. Crossed cells indicate infeasible solution characteristics.

different quality. Such sharing of representations occurs infrequently and is discussed below in Section 4.4.

The second criterion for unique representation is that each distinct solution must be represented no more than once. If this latter criterion is not met, then ants could construct the same solution while using different sets of solution characteristics (this is illustrated for a subset problem in Figure 1). Moreover, pheromones that represent distinct solutions multiple times consequently increase the size of solution space ants search in—even though the same number of distinct solutions are present—which can be an undesirable property for search algorithms. For many problems, the solution structure allows each solution to be represented in multiple ways. Due to constraints present in the problem, the distribution of these extra representations is likely to be nonuniform, biasing the search towards certain solutions. A bias that favors better solutions would be desirable. However, as the distribution of representations is unknowable for difficult combinatorial problems, unique representation of solutions is a safe, practical alternative. Notably, Gutjahr [23] states for GBAS that (p. 875) "... to each feasible solution, there corresponds (via Φ^{-1}) at least one walk . . . [in the construction graph]." This means that the pheromone it uses, which is associated with the links in these walks, is not guaranteed to have the property of unique representation.

It should be noted that with any pheromone representation ants are able to construct the same solution while encountering different sequences, and in some cases slightly different subsets, of the solution characteristics that describe that solution. For instance, in the TSP ants may construct the same solution while starting at different cities, thereby encountering a different sequence of solution characteristics (i.e., links) while never explicitly encountering the link from the last back to the first city, which may be explicitly visited by other ants producing the same solution. With higher order pheromone representations, ants may encounter different sets of higher order solution characteristics, as the contents of their respective partial solutions will often be different even if those partial solutions will eventually represent the same solution. Both cases are artifacts of the constructive approach and are not inherently problematic, as at each step ants are given the best available estimate of the learned utility of solution components. A problem arises when two or more solution characteristics from a given pheromone representation correspond to a single "true" solution characteristic in terms of a particular problem. For instance, using a $C \times P$ pheromone representation for a subset problem means that the solution characteristics $(i, 1), \ldots, (i, n)$ all represent the single solution characteristic "i is present in the solution." Yet with this representation only one of these is used in each ant's solution, spreading the pheromone for the important solution characteristic across multiple values. Consequently, each distinct solution also has multiple representations. Indeed, using the $C \times P$ pheromone for this problem, each solution of n items has $n!$ different representations.

Given that the pheromone is the mechanism for learning about features of good solutions in ACO, the property of unique representation is one of the most important determinants of a pheromone representation's utility for a given problem. Where the solution structure imposed by the constructive process allows for each solution to be represented only once, all alternative pheromones for that structure will have the property of unique representation. Where the solution structure used does allow for distinct solutions to be represented in different ways, pheromones based on how solutions are represented will also represent distinct solutions more than once. Even when alternative pheromones for a problem have the property of unique representation, it does not follow that they will be equally effective, an issue discussed in Section 6.

4.1 Unique Representation in Higher Order Representations

If a given first order pheromone representation has the uniqueness property for a particular problem, then any higher order representations generated from it will also have the uniqueness property on the same problem. This is formalized in Theorem 1. This principle holds for all nth order pheromone representations given solutions containing at least n first order solution characteristics; solutions with fewer than that do not possess groups of the required size.

Theorem 1: *Let M be an arbitrary first order pheromone representation. If M has the property of unique representation for a given problem, then solutions represented by at least $n \geq 2$ different pheromone values are also uniquely represented by any higher order representation M^n.*

Proof. Let M be an arbitrary first order pheromone representation such that M has the property of unique representation. Let $P_M \subset M$ be a set of solution characteristics from M that corresponds to a solution to the problem under consideration. Let $P_{M^n} = \{ (i_1, \ldots, i_n) \mid i_1; \ldots ; i_n \in P_M \, i_1 \prec \cdots \prec i_n\}$ be the set of all n-tuples of distinct elements from P_M, where $\prec$ is an arbitrary but fixed order imposed on the elements of M. There is only one such set of n-tuples for each P_M and n. From Definition 1 it can be seen that $P_{M^n} \subset M^n$. Therefore, each solution has exactly one representation in M^n. In view of Definition 2, M^n has the property of uniqueness.

Corollary 1: *Let M be an arbitrary first order pheromone representation such that M does not have the property of unique representation for a given problem. Provided that solutions are represented by at least n different pheromone values in M, any higher order representation M^n also does not possess the property of unique representation for that problem.*

4.2 Examples of Unique and Multiple Representation

This section examines alternative pheromone representations for three important COPs to illustrate the importance of the property of unique representation.

4.2.1 Subset Problems (Cost on Items)

Consider again subset problems where cost or profit is associated with the items included in the subset. This includes the MKP, SCP, SPP, and k-cardinality tree problem, among others. Intuitively, such problems should have a C pheromone representation, as the items placed in the subset are important solution characteristics. Indeed, TSP-like and other pheromone representations are regarded as inappropriate for these problems [26]. Nevertheless, they make an interesting case study in the analysis of inappropriate pheromone representations.

When applying a $C \times C$ pheromone representation to these problems, a solution characteristic (i, j) represents choosing one item i after choosing some other item j. It derives from the graph representation of a subset, where the nodes visited are the items included in the subset and pheromone is associated with the links in that graph, which are clearly meaningless for a subset problem. However, the definition of GBAS suggests that this pheromone is the one that it would use when applied to subset problems. If solutions are represented in this way, with an artificial start node (an item with zero weight and cost) from which all ants begin, each solution of n items is represented $n!$ times, with the pheromone associated with including a given item spread over k separate values, where k is the total number of items (excluding the start node) $n < k$. Figure 1 illustrates the use of this pheromone representation for a simple subset problem. As described above, a $C \times P$ pheromone creates the same number of extra representations for solutions to these problems.

The intuitive pheromone choice C satisfies the requirements of unique representation, as any solution may be represented, yet each solution is represented by at most one subset of solution characteristics.

4.2.2 Graph Coloring Problem

In the graph coloring problem, although solution components represent the assignment of specific colors to nodes, distinct solutions are described by the groups of like-colored nodes. That is, specific color information is discarded. Thus, for a given k-coloring of a graph, there are $k! - 1$ other colorings that represent the same solution, produced by swapping the actual colors between color groups. Clearly, then, any pheromone representation that includes specific color information may represent distinct solutions multiple times. This is the case if using a $C_1 \times C_2$ pheromone, where C_1 is the set of nodes and C_2 is the set of colors. Each of the $k! - 1$ alternative representations of each distinct solution has a different corresponding set of solution characteristics (i.e., representation) in the pheromone. Furthermore, such a pheromone may mislead ants by attracting them to make node-color assignments that, being parts of two different representations of the same distinct solution, produce a poorer solution when combined.

Similar problems occur if either of the second order representations $X \times C_1 \times C_2; (C_1 \times C_2)^2$ or $X \times C_1 \times C_2; (C_1)^2 \times C_2$ is used. The former stores pheromone between all pairs of node-color assignments, while the latter only stores pheromone for like-colored pairs of nodes. While these representations capture interdependences between adjacent nodes so that poor combinations of node-color assignments are less likely, each solution is still represented multiple times. For instance, consider the solution characteristic (i, j, k) taken from the latter pheromone, which is equivalent to (i, j, k') if the colors k and k' are swapped. As described in Section 3, these representations may be simplified by discarding specific color information to produce a grouping pheromone for this problem. An $X \times C_1 \times C_2; (C_1)^2$ grouping pheromone, applied to this problem, does have the property of uniqueness, as it directly models color groups.

4.2.3 Permutation Scheduling Problems

The solution to a number of scheduling problems, collectively called permutation scheduling problems, may be formulated as a permutation of a set of operations, subject to some partial order. This group includes the single machine total tardiness problem (and variants) and the flow shop, job shop (JSP), and open shop (OSP) problems. In some problems, operations are grouped into jobs. Due to precedence constraints between some pairs of operations within each job, the completion time for each operation, and hence its contribution to the schedule's overall cost, will depend on differing subsets of the operations that precede it. Consequently, different permutations may actually represent the same solution. Early ACO algorithms for these problems used a TSP-like $C \times C$ pheromone representation (C is the set of operations), which has the drawback that an artificial start node representing an empty schedule must be used, as solutions to these problems are not cyclic like those to the TSP (e.g., [11]). More recent algorithms have used a $C \times P$ pheromone (P is the absolute position of operations in the sequence), which does not require an artificial start node. Both

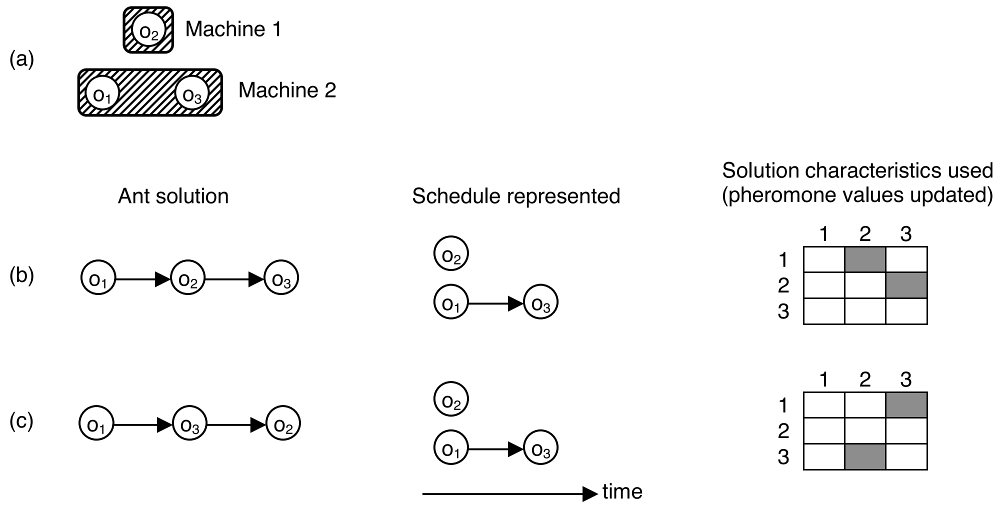

Figure 2. A small OSP-style permutation scheduling problem using the $C \times C$ pheromone representation. The artificial start point $\varnothing$ is not shown for brevity. (a) o_1 and o_3 must be processed on the same machine, while o_2 requires a different machine. (b) An ant's solution, the schedule it represents, and the solution characteristics used. (c) Another ant's solution, producing the same schedule but using different solution characteristics.

of these represent permutations rather than precedence relationships between operations and hence can represent solutions multiple times.

Consider the application of a $C \times C$ pheromone to the trivial scheduling problem depicted in Figure 2. In this example, the completion time of o_3 may be affected by the completion time of o_1 (and vice versa), while o_2 is independent of o_1 and o_3. As a result, the operation sequence $\langle o_1, o_2, o_3 \rangle$ is equivalent to $\langle o_1, o_3, o_2 \rangle$, even though different solution characteristics from $C \times C$ are involved in each. In empirical studies of the $C \times C$ pheromone representation, Blum and Sampels [7, 8] found that it introduced a bias into the search, such that in more constrained scheduling problems like the JSP, poorer solutions could be reinforced more than better solutions.

To counteract some empirically observed problems with the $C \times P$ pheromone, Merkle and Middendorf [29, 30] suggest two alternative schemes for interpreting and updating pheromone values, called summation evaluation [29] and relative pheromone evaluation [30]. In terms of the notation introduced in Section 3, both of these approaches would be described as $C \times P$; $C \times P$, because the pheromone associated with placing an operation at the current location in a solution vector is derived from several pheromone values that describe the utility of placing that operation at different positions in the vector. Clearly, the notation is not perfectly suited to describing these uses of pheromone, as they use pheromone values associated with taking one action to influence the decision to take a different action. While these improve the results of using this pheromone representation on some scheduling problems, they still do not directly model the precedence relationships that uniquely describe solutions and so may represent solutions multiple times.

Blum and Sampels [7] describe an alternative pheromone representation, $X \times C$; C^2 (preceding), referred to as *learning-of-relations* pheromone, where solution characteristics $(i, j) \in C^2$ represent scheduling operation i somewhere before operation j. This pheromone representation is intended to identify precedence relations between related operations, where two operations are related if the scheduling of either one can have a direct effect on the start time of the other (excluding precedence relations required by the problem's constraints). For instance, in the case of the JSP, operations that require the same machine will affect each other's start time and are therefore related. The underlying C^2 pheromone representation allows for only one representation of each solution, as it directly models the precedence relations between all related operations and therefore has the property of unique representation. As reported in Section 1.1, it outperforms the other commonly used pheromone representations for these problems, including $C \times P$ with summation evaluation.

In summary, unique representation of solutions in pheromones is important for the successful application of ACO. Without it, ants explore enlarged search spaces with conflicting guidance from different pheromone patterns for otherwise identical solutions. The next section describes an alternative way of characterizing pheromone representations based on their size and discusses the implications for choosing between alternative pheromones.

4.3 Pheromone Representation Size and Unique Representation

Pheromones that lack the property of unique representation are most directly identified by examining the number of times they represent individual solutions. In this way direct comparisons may be made between alternative pheromone representations. A related approach considers the total number of solution characteristics described by alternative pheromone representations. Given that alternative pheromones for the same problem represent the same number of distinct solutions, and that each solution characteristic corresponds to a decision variable plus some value, a larger set of solution characteristics corresponds to a larger number of ways to represent each solution. More precisely, the larger the ratio between the total number of solution characteristics and the number of solution characteristics used per solution, the more likely it is that pheromone does not have the property of unique representation. However, the number of solution characteristics (i.e., values assigned to decision variables) used per solution is similar across alternative pheromone representations, because they must make similar numbers of decisions when constructing solutions. Thus, we consider only the total number of solution characteristics hereafter.

As larger sets of solution characteristics are more likely to allow for multiple representations of solutions, a *minimal* set of solution characteristics is a sufficient condition for possessing the property of unique representation. By *minimal* it is meant that there is no smaller set of solution characteristics that may still represent each solution separately. If this latter criterion were excluded, the minimal set of solution characteristics would belong to the pheromone representation with the single solution characteristic that a solution is feasible, which clearly possesses the property of unique representation, yet is singularly unhelpful in learning how to construct good solutions. Once a minimal set of solution characteristics is identified, creating higher order versions of it using Definition 1 or using a single pheromone value for groups of solution characteristics (discussed in the next section) will still produce pheromones with the property of unique representation, but with different learning properties from the original set.

While a minimal set of solution characteristics is sufficient for a pheromone to have the property of unique representation, it is not a necessary condition for all problems. For example, with an n operation SMTTP, $C \times C$ and $C \times P$ pheromones describe $n(n-1)$ and n^2 solution characteristics in total, respectively, of which n are used to describe a single solution. This is somewhat larger than Blum and Sampels' [7] learning-of-relations pheromone [$X \times C$; C^2 (preceding)], which only describes $n(n-1)$ solution characteristics in total, with $n(n-1)/2$ used for each solution. Yet all three pheromones have the property of unique representation for this problem. This does not entail that these pheromones will produce the same behavior when used in otherwise equivalent ACO algorithms, simply that there is only one way to represent each distinct solution in each of them. In more complex shop scheduling problems with multiple jobs and machines, only the learning-of-relations pheromone maintains the property of unique representation.

Section 6 shows how the automated pheromone selection scheme we propose ensures a minimal set of solution characteristics (or some higher order version thereof), thereby producing a pheromone representation with the property of unique representation for a given problem.

The next subsection explores issues related to pheromone representations that, while representing all solutions exactly once, share representations between distinct solutions.

4.4 Shared Representations in Pheromones

When a single set of solution characteristics corresponds to two or more distinct solutions, those solutions may be said to share a representation. In effect, it is the reverse of having multiple representations.[6] However, examination of a wide range of pheromone-representation–problem combinations suggests that it is far less common for multiple solutions to share one representation than for a single solution to have multiple representations. Furthermore, sharing of representations is not necessarily undesirable.

It is possible to contrive unnatural pheromone representations for a variety of problems that exhibit shared representations. For instance, consider a pheromone representation for a subset problem that arbitrarily groups items into pairs, with a single pheromone value to indicate if either item in each pair should be included in solutions. Such a representation would make use of a set of virtual components C', where $C \mapsto C'$ and $|C'| = |C|/2$. Similar pheromone representations may be contrived for other problems by having small groups of solution components where each group has a single pheromone value. Clearly, such representations are inappropriate, as they fail to distinguish between different solutions. However, these examples are highly artificial and unlikely to eventuate in the typical application of ACO. Of more interest is the small number of pheromone representations with this property that can appear as a result of plausible design decisions. Two of these are considered here.

4.4.1 Aircraft Landing Problem

The static single runway aircraft landing problem involves the allocation of landing times to planes such that allocated times are within each plane's landing window while minimum separation times between different types of aircraft are respected [37]. Each plane has a preferred (most economical) landing time, and the aim of the problem is to minimize the difference between actual and preferred landing times, with different penalty rates for earliness and lateness. If the problem is modeled as the assignment of times to planes, then an intuitive choice of pheromone is $C_1 \times C_2$, where C_1 is the set of planes and C_2 is the set of landing times. However, given that time windows differ between planes and can be of different lengths, such a representation contains many solution characteristics that can never be used.[7] Further, unlike many other assignment problems where the number of items (in this problem, planes) is equal to or greater than the number of resources (time slots), in this problem $|C_1| \ll |C_2|$.

To create a more manageable pheromone representation, Randall [37] divides each plane's time window into a fixed number k of contiguous regions. While solution components are from $C_1 \times C_2$, the solution characteristics are from $C_1 \times C'_2$, where $C_1 \times C_2 \mapsto C'_2$. Hence, solutions with assigned landing times that fall within the same regions will map to the same set of solution characteristics. This kind of pheromone sharing is possible due to the nature of the resources (i.e., time slots) in this problem. Unlike problems such as the GAP, where resources have quite distinct identities, resources in this problem vary only slightly from one to the next. Furthermore, given the large range of time slots available, it may even be advantageous to group closely related time slots in this manner. If the pheromone were associated with individual time slots, each time slot would have to be assigned to the same plane a number of times before pheromone levels would be sufficiently high to seriously affect ant behavior. Aggregating adjacent time slots increases the likelihood of any one of them being chosen. If desired, the number of regions could be increased as the search progresses to differentiate between neighboring time slots.

Thus, for the single runway aircraft landing problem, modeled as the assignment of time slots to planes, using a pheromone that shares representations may be a good practical choice. If the problem is modeled so that solutions determine the landing order of planes only, with actual landing times to be assigned by a subordinate heuristic, an alternative pheromone that does not share

6 While it would be possible to contrive a pheromone representation that exhibits multiple shared representations, it is improbable that such pheromones would be developed in the normal course of applying ACO, so they are not considered further.

7 This is the case if the representation is implemented as a matrix.

representations may be more appropriate, such as Blum and Sampels' [7] learning-of-relations pheromone.

4.4.2 Car Sequencing Problem

The car sequencing problem is a common problem in the car manufacturing industry [40]. The aim of the variant considered here is that cars of different models are placed in a production sequence such that the separation penalty between cars of the same model is minimized.[8] Each model has a fixed number of cars. One way this can be modeled is as the allocation of sequence positions to different models. This shares similarities with graph coloring, in that the solution cost relates to pairs of items (i.e., sequence positions) assigned to each group (i.e., model); yet it also has features of a typical assignment problem, in that the separation penalty depends on what model a sequence position is assigned to. A highly appropriate pheromone representation for this problem model is developed in Section 6. However, in order to illustrate potential problems with shared solution representations, only a pheromone for the graph coloring aspects is considered here. An $X \times C_1 \times C_2$; $(C_1)^2$ pheromone captures how separation penalties are allocated within models (groups). However, while each solution is represented no more than once in this pheromone, several solutions share their representation with other solutions. This is because, unlike graph coloring, the groups in this problem have separate identities, evidenced by the different penalties associated with each model. This pheromone representation is therefore inappropriate, as very different solutions may be represented by the same set of solution characteristics.

Whether or not to share pheromone representations between different solutions seems to be a practical consideration depending on the problem in question. The automated system described in Section 6 does not currently support pheromones that share representations between solutions.

5 Parsimony in Pheromone Representations

Another desirable property of pheromone representations is parsimony, which means that a representation contains just enough information to correctly represent key solution characteristics, and no more. Generally, this relates to the use of a second order representation when a first order representation will suffice. The following examples illustrate the role of parsimony in pheromone representations.

In subset problems such as the multiple knapsack and k-cardinality tree problems, it is possible to use the second order pheromone representation $X \times C$; C^2 (copresent). Due to Theorem 1, this second order representation has the property of unique representation. Additionally, the extra information provided may promote better learning about good solutions to the problem in question. However, in a comparative study on the k-cardinality tree problem, Blum [4] found that it was not as effective as a C pheromone representation. It was suggested that this is due to the increased computational overhead of using a more complicated pheromone representation, which results in fewer solutions produced given the same amount of time (as was the case in this study). This suggests that if a simpler pheromone representation adequately models a problem, it is unnecessary to use a higher order (commonly, a second order) pheromone representation.

The same is likely to be true of the higher order pheromone representations for assignment problems, such as $X \times C_1 \times C_2$; $(C_1 \times C_2)^2$, which models pairs of assignments from $C_1 \times C_2$ being copresent in a solution. This pheromone also has the property of unique representation, as solutions to these problems are uniquely defined by the assignments from which they are assembled. However, unless the typical assignment pheromone is not sufficient to capture those solution characteristics that are important in a problem, parsimony suggests that this second order representation should be avoided used in favor of its first order counterpart.

8 The car sequencing problem 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 [22].

While not having as large an effect on ACO's effectiveness as a pheromone representation lacking uniqueness, the use of a representation lacking parsimony may affect an ACO algorithm's performance. The computational overhead required to process a representation with extra information must be weighed against any potential benefits in solution quality.

6 Systematically Determining Appropriate Pheromone Representations

A pheromone representation that both uniquely represents solutions and is parsimonious may not adequately describe solution characteristics that affect solution cost. Consider a subset problem characterized by an objective function where cost is related to some relationship between the pairs of items in the subset, with an objective function of the form $\Sigma_i \Sigma_{j \neq i} f(x(i), x(j))$. The maximum clique (MCP) and N-queens (NQP) problems can be formulated in this way. As with simpler subset problems, the $C \times C$ and $C \times P$ pheromone representations are inappropriate, as they allow otherwise identical solutions to be represented more than once. A C pheromone representation has the property of unique representation, and is also parsimonious in view of simplicity. However, the objective function suggests that the impact of including one item in the subset is related to all other items in the subset, which is modeled by the second order pheromone representation $X \times C$; C^2 (copresent).

This example reveals a more general principle concerning problems and the most appropriate choice of pheromone. While multiple solution representations and lack of parsimony are undesirable, they are actually indications that a pheromone representation fails to adequately model those features of solutions that directly affect solution cost. Ants construct solutions from a set of solution components independent of the pheromone representation. Depending on the problem, each distinct solution may be described by different sets or arrangements of solution components. This is the case in the graph coloring problem, where solution components represent node-color assignments while solutions are only uniquely described in terms of color groups. Consider an alternative solution representation based on how various parts of the solution (i.e., solution components or parts of solution components) directly affect the solution cost. In effect, this involves defining the decision variables of the problem so that a change in the value of any of them results in a change in the solution represented and hence the cost.[9] As such a set of decision variables ignores how solutions are represented (except where this has an effect on the solution cost), it follows that it represents solutions exactly once, and also provides a minimal set of decision-variable–value pairs. In order to adequately model those features of solutions that directly affect the solution cost, and hence produce a parsimonious pheromone representation that uniquely represents solutions, solution characteristics must be chosen such that the decision variables they define have this property. This information can be derived from a problem's objective function and in some cases certain constraints.

The objective function for many common COPs consists of a summation over a number of terms. Each of these terms may be considered as a solution characteristic. The problem of deriving an appropriate pheromone representation then becomes that of identifying the nature of these terms. To facilitate this task, a suitable modeling language must be used that helps reveal how different parts of the solution relate to each other. Hence, a 0-1 integer linear programming formulation would be inappropriate as the true nature of a problem is often difficult to discern given the numerous 0-1 variables and constraints involved.

Our algorithm for automatically selecting pheromone representations requires that the problem model has already been parsed and that the nature of the entities (i.e., components) added at

9 It is conceivable that in using such a representation two apparently different solutions will have the same objective cost. However, unless they can be shown to be completely equivalent, taking into account factors other than their cost, they should be considered to be distinct. For instance, two solutions to the TSP may have the same tour length despite visiting cities in different orders, and would then be considered as independent solutions.

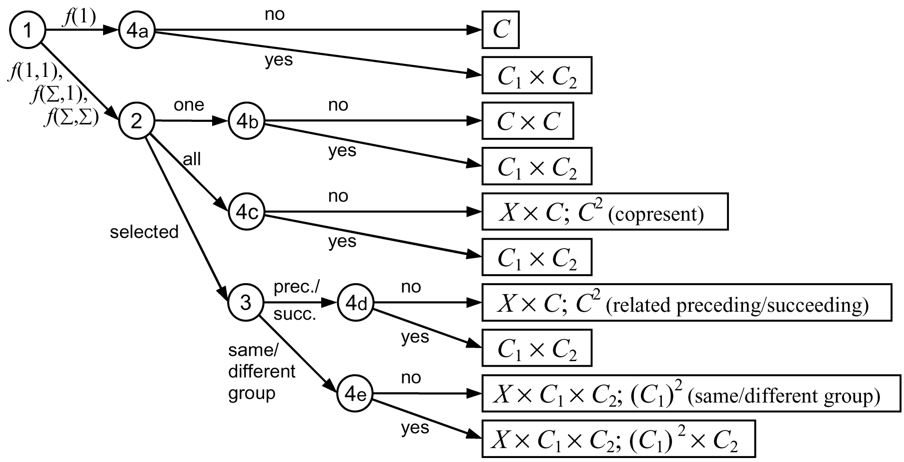

Figure 3. Decision tree for automated selection of pheromone representations.

each step be known, even though their impact on the solution cost is not. Hence, the algorithm identifies the nature of the solution components and/or characteristics in the problem. The algorithm has been developed by studying a wide range of commonly occurring COPs to identify recurring themes in the objective function and what these indicate about cost contributors. This is not the only possible algorithm for this task, but appears well suited to those problems examined. The algorithm follows a decision tree (see Figure 3) that uses the four questions outlined below. For each question, the possible answers are given, followed by an explanation of the reason for the question.

Note that $f(1)$ represents a function of one part of the solution while Σ represents an aggregate of many parts of the solution. For example, the SMTTP has an objective in which each operation's contribution to the solution cost is related to $\Sigma_j^{j<i} p(x(j)) - d(x(i))$, where $x(i)$ is the ith operation in the permutation, and $p(k)$ and $d(k)$ represent the processing time and due date of operation k, respectively. This corresponds to $f(\Sigma, 1)$. The number partitioning problem (NPP) [25] has a single term of the form $\Sigma_i x(1, i) - \Sigma_j x(2, j)$, where $x(i, j)$ is the jth number in partition i, which matches $f(\Sigma, \Sigma)$. The algorithm is currently limited to objective functions consisting of no more than two parts.

1. How many parts of the solution are used in each term in the objective?

   Possible answers: $f(1)$, $f(1, 1)$, $f(\Sigma, 1)$, $f(\Sigma, \Sigma)$.

   Rationale: To know if the cost is related to individual parts of the solution or to some relationship between parts. $f(1)$ restricts the number of feasible pheromone representations, while the others require further examination.

2. If each part is related to another part of the solution, how many other parts is it related to?

   Possible answers: *one*, *selected* (and more than one), *all*.

   Rationale: To determine the scope of the relationship(s) between parts. The answer *one* restricts the number of feasible pheromone representations, while the other two require further examination. *Selected* indicates that each part is related to *some* of the other parts, but not all. For instance, in the JSP the contribution to the solution cost of a single operation is related to what other operations that use the same machine have been scheduled before it, which is a subset of all operations, since JSPs typically involve multiple machines. In the SMTTP an operation's contribution to the solution cost is related to all operations that precede it, but none that succeed it, which also matches the answer *selected*.

3. If each part is related to a selected group of other parts, what identifies them?

   Possible answers: *all preceding/succeeding, selected preceding/succeeding, assigned same group/resource, assigned different group/resource.*

   Rationale: To differentiate between problems where the relative order of components is important (first two answers) and those where group assignment is important (second two answers). In an implementation of the system, the two answers corresponding to each kind of relationship (ordered versus group assignment) would be used to make minor changes to which parts of the pheromone representation would be used in decisions and updated by solutions. For instance, the SMTTP would match *all preceding*, while the JSP would match *selected preceding*.

4. If the part referenced represents an assignment, is the resource assigned used separately in the objective function?

   Possible answers: *yes, no* (or solution component is not an assignment).

   Rationale: If solution components represent assignments and each term in the objective is some function of both the item and the resource to which it is assigned, then it is likely that it is the assignment that is most important, rather than any other relationship(s) in the problem. The exception to this rule is when dealing with a problem in which group membership is important as well as *which* group items are assigned to, in which case both aspects must be reflected in the pheromone representation.

A simple decision tree is depicted in Figure 3. Note that Question 4 appears at the end of most branches and in all but one case overrides any other characteristics of the problem. This is because if the objective is, at some level, a function of individual assignments, then associating pheromone with these will ensure that the representation has the property of unique representation while still representing an important contributor to the solution cost. To illustrate how the algorithm may be applied, consider the following four problems. Objective functions have been simplified by removing the bounds on summations.

TSP Objective: $\Sigma_i\, d(x(i), x(predecessor(i)))$, where $x(i)$ is the ith city in the solution x, $d(i, j)$ is the distance between cities i and j, and $predecessor(i)$ returns the preceding position to i, which is the last position in the solution if i is the first solution position, $i - 1$ otherwise.

- Question 1: $f(1, 1)$, goto Question 2;
- Question 2: *one*, goto Question 4b;
- Question 4b: *no,* select $C \times C$ (adjacent pairs) pheromone.

QAP Objective: $\Sigma_i\, \Sigma_j\, a(i, j) \cdot b(x(i), x(j))$, where $x(i)$ is the facility assigned to location i, $a(i, j)$ is the distance between locations i and j and $b(k, l)$ is the amount of flow between facilities k and l.

- Question 1: $f(1, 1)$, goto Question 2;
- Question 2: *all*, goto Question 4c;
- Question 4c: *yes*, select $C_1 \times C_2$ pheromone.

GAP Objective: $\Sigma_i \Sigma_j a(x(i, j), i)$, where $x(i, j)$ is the jth task assigned to agent i, and $a(k, l)$ is the cost of assigning task k to agent l.

- Question 1: $f(1)$, goto Question 4a;
- Question 4a: *yes*, select $C_1 \times C_2$ pheromone.

Car sequencing problem (CSeqP) A description of this problem appears in Section 4.4. In this formulation, sequence positions are allocated to a fixed number of cars within each model.

Objective: $\Sigma_i \Sigma_j \Sigma_k P(|x(i, k) - x(i, j)|, i)$, where $x(i, j)$ is the position assigned to the jth car of model i,[10] and $P(i, j)$ is the penalty for cars of model j separated by i positions.

- Question 1: $f(1, 1)$, goto Question 2;
- Question 2: *selected*, goto Question 3;
- Question 3: *same group*, goto Question 4e;
- Question 4e: *yes*, select $X \times C_1 \times C_2$; $(C_1)^2 \times C_2$ pheromone.

Note that in addition to the decision tree, an implementation of the system must perform some other processing to slightly tailor the chosen pheromone representation to match a particular problem. For instance, with the TSP, the system must be able to recognize that the *predecessor*() function relates adjacent pairs, rather than pairs separated by some other distance. In problems such as the JSP, the system must be able to determine which other operations are capable of affecting each operation's start time, information obtainable from the problem constraints and objective.

The results from applying the algorithm to a range of COPs are presented in Table 2. With the exception of the SMTTP, the suggested pheromone representation is the best known pheromone for each problem to which ACO has been applied in the literature. The suggested pheromone for the SMTTP has not been used with that problem, so no comparison can be made.

7 Concluding Remarks

As the range of application of ACO has grown, so too has the variety of solution and pheromone representations, at times departing considerably from the original AS for the TSP and the natural metaphor on which it is based. Given that the pheromone is the mechanism for learning about good features of solutions in ACO, the utility of a pheromone representation may be largely determined by whether or not it uniquely represents solutions. Pheromone representations are generally chosen in an ad hoc manner, which has in some instances produced representations that do not uniquely represent solutions. This increases the effective size of the search space and may potentially mislead the search process as to the true learned value of solutions. Pheromone representations with too much information can slow down the search process, although in some circumstances they may produce improvements in solution quality that outweigh losses in speed. This issue is currently being further investigated. Nevertheless, many intuitive pheromone choices have been highly effective in a number of instances, suggesting that deriving pheromone representations by the systematic analysis of a problem, especially its objective function, may yield improved results. Moreover, deriving a pheromone representation in this way ensures solutions are represented both uniquely and parsimoniously, while also focusing learning on those characteristics of solutions that most directly contribute to cost.

We have described a system for the automated selection of pheromone representations that may be applied to a wide range of COPs. In general, this system's predictions correlate with the

10 Although the order of car positions within each model group is not important, it is convenient to represent each model's set of positions as a list.

Table 2. Suggested pheromone representations for common COPs using the new selection system. The problem name abbreviations not previously defined are as follows: VRP = vehicle routing, LOP = linear ordering, GPP = graph partitioning. PAP = processor allocation. Other abbreviations used: sel'd = selected, prec. = preceding, group = same group, diff. = different group. References are given where ACO has been applied to that problem using the suggested pheromone.

	Decision process				Suggested	
Problem	Q1	Q2	Q3	Q4	pheromone	Reference
MKP	$f(1)$	—	—	no	C	[26]
SCP	$f(1)$	—	—	no	C	[21, 35]
SPP	$f(1)$	—	—	no	C	
GAP	$f(1)$	—	—	yes	$C_1 \times C_2$	[27]
TSP	$f(1, 1)$	one	—	no	$C \times C$	[16]
VRP	$f(1, 1)$	one	—	no	$C \times C$	[9]
NQP	$f(1, 1)$	all	—	no	$X \times C;C^2$ (copresent)	
MCP	$f(1, 1)$	all	—	no	$X \times C;\ C^2$ (copresent)	[20]
QAP	$f(1, 1)$	all	—	yes	$C_1 \times C_2$	[43]
FAP	$f(1, 1)$	all	—	yes	$C_1 \times C_2$	[28]
SMTTP	$f(\Sigma, 1)$	sel'd	prec.	no	$X \times C;\ C^2$ (all prec.)	
JSP	$f(\Sigma, 1)$	sel'd	prec.	no	$X \times C;\ C^2$ (sel'd prec.)	
LOP	$f(\Sigma, 1)$	sel'd	prec.	no	$X \times C;\ C^2$ (all prec.)	
GRAPH	$f(1, 1)$	sel'd	group	no	$X \times C_1 \times C_2;\ (C_1)^2$ (group)	[12]
CSP	$f(1, 1)$	sel'd	group	no	$X \times C_1 \times C_2;\ (C_1)^2$ (group)	[18]
BIN	$f(1, 1)$	sel'd	group	no	$X \times C_1 \times C_2;\ (C_1)^2$ (group)	[18]
GPP	$f(1, 1)$	sel'd	diff.	no	$X \times C_1 \times C_2;\ (C_1)^2$ (diff.)	
PAP	$f(1, 1)$	sel'd	diff.	no	$X \times C_1 \times C_2;\ (C_1)^2$ (diff.)	

Table 2. (continued)

Problem	Decision process Q1	Q2	Q3	Q4	Suggested pheromone	Reference
NPP	$f(\Sigma, \Sigma)$	sel'd	diff.	no	$X \times C_1 \times C_2$; $(C_1)^2$ (diff.)	
CSeqP	$f(1, 1)$	sel'd	group	yes	$X \times C_1 \times C_2$; ${C_1}^2 \times C_2$	

best-known pheromone representations for problems described in the literature. Initial results from our empirical studies with alternative pheromones for the TSP, JSP, OSP, GAP, and QAP support the system's pheromone choices for these problems. The system also makes predictions about problems that ACO has not yet been applied to.

Additionally, we have introduced a notation for the formal description of pheromone representations. There are a number of complex problems for which this system is currently unable to derive a single most appropriate pheromone. These include multiple objective problems such as the SMTTP with changeover costs [24], and multiple task problems, like the p-hub median class of problems [19], where solutions are built in two separate, but interacting, stages. To handle these problems the hard decision tree will likely need to be softened to allow several pheromone representations to be suggested.

A limitation of the current study is that the interaction between a pheromone representation and the particular features of a given ACO algorithm is not considered. ACO algorithms do not, in general, rely solely on pheromone information to guide the search process, making use of heuristic estimates of solution components' values and also manipulating pheromone information in different ways.

Furthermore, the nature of the constructive process used by an ACO algorithm may introduce a bias into the search that can affect the utility of a pheromone, even if it represents solutions uniquely [34]. Different aspects of this issue have only recently been investigated in detail, by Blum [5] and also by the authors [34]. Future work will integrate these two approaches to understanding and predicting pheromone performance with the ideas presented in this article. As part of this integration, the performance of the suggested pheromone representations will be compared against other pheromone representations on a range of problems.

References

1. Abramson, D., & Randall, M. (1999). A simulated annealing code for general integer linear programs. *Annals of Operations Research*, *86*, 3–21.

2. Bauer, A., Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). *Applying ant colony optimization to solve the single machine total tardiness problem* (Technical report No. 42). Vienna, Austria: Vienna University of Economics and Business Administration.

3. Birattari, M., Di Caro, G., & Dorigo, M. (2002). Toward the formal foundation of ant programming. In M. Dorigo, G. Di Caro, & M. Sampels (Eds.), *Proceedings of the Third International Workshop on Ant Algorithms, ANTS 2002*, *Brussels, Belgium* (pp. 188–201). Berlin: Springer-Verlag.

4. Blum, C. (2002). Ant colony optimization for the edge-weighted k-cardinality tree problem. In W. Langdon (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), New York, USA* (pp. 27–34). San Francisco: Morgan Kaufman.

5. Blum, C. (2004). *Theoretical and practical aspects of ant colony optimization*. Volume 282 of Dissertations in Artificial Intelligence. Nieuwe Hemweg, The Netherlands: IOS Press.

6. Blum, C., & Dorigo, M. (2003). The hyper-cube framework for ant colony optimization. IEEE *Transactions on Systems, Man and Cybernetics—Part B*, *34*, 1161–1172.

7. Blum, C., & Sampels, M. (2002). Ant colony optimization for FOP shop scheduling: A case study on different pheromone representations. In *Proceedings of the 2002 Congress on Evolutionary Computation* (pp. 1558–1563).

8. Blum, C., & Sampels, M. (2002). When model bias is stronger than selection pressure. In J. M. Guervós, P. Adamidis, H. G. Beyer, J. L. Fernández-Villacañas, & H. P. Schwefel (Eds.), *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN2002)* (pp. 893–902). Berlin: Springer-Verlag.

9. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, *89*, 319–328.

10. Burke, E. K., Hart, E., Kendall, G., Newall, J., Ross, P., & Schulenburg, S. (2002). Hyper-heuristics: An emerging direction in modern search technology. In F. Glover, & G. Kochenberger (Eds.), *Handbook of meta-heuristics* (pp. 457–474). Boston: Kluwer Academic Publishers.

11. Colorni, A., Dorigo, M., Maniezzo, V., & Trubian, M. (1994). Ant system for job-shop scheduling. *Belgian Journal of Operations Research*, *Statistics and Computer Science*, *34*, 39–53.

12. Costa, D., & Hertz, A. (1997). Ants can colour graphs. *Journal of the Operational Research Society*, *48*, 295–305.

13. Di Caro, G., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, *9*, 317–365.

14. Dorigo, M., Bonabeau, E., & Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Generation Computer Systems*, *16*, 851–871.

15. Dorigo, M., & Di Caro, G. (1999). The ant colony optimization metaheuristic. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 11–32). London: McGraw-Hill.

16. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics—Part B*, *26*, 1–13.

17. Dorigo, M., & Stützle, T. (2002). The ant colony optimization metaheuristic: Algorithms, applications and advances. In F. Glover, & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 251–285). Boston: Kluwer Academic.

18. Ducatelle, F., & Levine, J. (2004). Ant colony optimisation and local search for bin packing and cutting stock problems. *Journal of the Operational Research Society*, *55*, 705–716.

19. Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, *4*, 139–154.

20. Fenet, S., & Solnon, C. (2003). Searching for maximum cliques with ant colony optimization. In S. Cagnoni, J. Romero Cardalda, D. Corne, J. Gottlieb, A. Guillot, E. Hart, C. Johnson, E. Marchiori, J. A. Meyer, M. Middendorf, & G. Raidl (Eds.), *Proceedings of the EvoWorkshops 2003, Essex, UK* (pp. 236–245). Berlin: Springer-Verlag.

21. Fiorenzo Catalamo, M. S., & Malucelli, F. (2001). Parallel randomized heuristics for the set covering problem. In M. Paprzycki, L. Tarricone, & L. T. Yang (Eds.), *Practical parallel computing* (pp. 113–132). Hauppause, NY: Nova Science Publishers.

22. Gottlieb, J., Puchta, M., & Solnon, C. (2003). A study of greedy, local search, and ant colony optimization approaches to car sequencing problems. In S. Cagnoni, J. Romero Cardalda, D. Corne, J. Gottlieb, A. Guillot, E. Hart, C. Johnson, E. Marchiori, J. A. Meyer, M. Middendorf, & G. Raidl (Eds.), *Proceedings of the EvoWorkshops 2003*, *Essex, UK* (pp. 246–257). Berlin: Springer-Verlag.

23. Gutjahr, W. J. (2000). A graph-based ant system and its convergence. *Future Generation Computer Systems*, *16*, 873–888.

24. Iredi, S., Merkle, D., & Middendorf, M. (2001). Bi-criterion optimization with multicolony ant algorithms. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01), Zurich* (pp. 359–372). Berlin: Springer-Verlag.

25. Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1991). Optimization by simulated annealing: An experimental evaluation; Part ii, graph coloring and number partitioning. *Operations Research*, *39*, 378–405.

26. Leguizamón, G., & Michalewicz, Z. (1999). A new version of ant system for subset problems. In *Proceedings of the 1999 Congress on Evolutionary Computation* (pp. 1459–1464). Piscataway, NJ: IEEE.

27. Lourenço, H. R., & Serra, D. (2002). Adaptive search heuristics for the generalized assignment problem. *Mathware and Soft Computing*, *9*, 209–234.

28. Maniezzo, V., & Carbonaro, A. (2000). An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, *16*, 927–935.

29. Merkle, D., & Middendorf, M. (2000). An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In *Proceedings of the EvoWorkshops 2000* (pp. 287–296). Berlin: Springer-Verlag.

30. Merkle, D., & Middendorf, M. (2002). Ant colony optimization with the relative pheromone evaluation method. In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, & G. Raidl (Eds.), *Proceedings of the EvoWorkshops 2002*, *Kinsale, Ireland,* (pp. 321–329). Berlin: Springer-Verlag.

31. Merkle, D., Middendorf, M., & Schmeck, H. (2000). Ant colony optimization for resource-constrained project scheduling. In D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, & H. G. Beyer (Eds.), *Proceedings of the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, NV, USA* (pp. 893–900). San Francisco: Morgan Kaufman.

32. Michel, R., & Middendorf, M. (1999). An ant system for the shortest common supersequence problem. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 51–61). London: McGraw-Hill.

33. Montgomery, J., Randall, M., & Hendtlass, T. (2003). Automated selection of appropriate pheromone representations in ant colony optimisation. In H. A. Abbass, & J. Wiles (Eds.), *Proceedings of the First Australian Conference on Artificial Life (ACAL03)*, *Canberra, Australia* (pp. 170–184). University of New South Wales.

34. Montgomery, J., Randall, M., & Hendtlass, T. (2004). Search bias in constructive metaheuristics and implications for ant colony optimisation. In *Proceedings of the Fourth International Workshop on Ant Algorithms, ANTS 2004, Brussels, Belgium*. Berlin: Springer-Verlag.

35. Rahoual, M., Hadji, R., & Bachelet, V. (2002). Parallel ant system for the set covering problem. In M. Dorigo, G. Di Caro, & M. Sampels (Eds.), *Proceedings of the Third International Workshop on Ant Algorithms, ANTS 2002, Brussels, Belgium* (pp. 262–267). Berlin: Springer-Verlag.

36. Randall, M. (2002). A general framework for constructive meta-heuristics. In E. Kozan, & A. Ohuchi (Eds.), *Operations research/management science at work* (pp. 111–128). Boston: Kluwer Academic.

37. Randall, M. (2002). Scheduling aircraft landings using ant colony optimisation. In *Proceedings of the Sixth IASTED International Conference on Artificial Intelligence and Soft Computing, Banff, Alberta, Canada* (pp. 129–133).

38. Randall, M., & Abramson, D. (2001). A general meta-heuristic based solver for combinatorial optimisation problems. *Journal of Computational Optimization and Applications*, *20*, 185–210.

39. Roli, A., Blum, C., & Dorigo, M. (2001). ACO for maximal constraint satisfaction problems. In *Proceedings of the Fourth Metaheuristics International Conference, Porto, Portugal* (pp. 187–192).

40. Smith, K., Palaniswami, M., & Krishnamoorthy, M. (1996). A hybrid neural network approach to combinatorial optimisation. *Computers and Operations Research*, *73*, 501–508.

41. Socha, K., Knowles, J., & Sampels, M. (2002). A MAX-MIN ant system for the university course timetabling problem. In M. Dorigo,G. Di Caro, & M. Sampels (Eds.), *Proceedings of the Third International Workshop on Ant Algorithms, ANTS 2002*, *Brussels, Belgium* (pp. 1–13). Berlin: Springer-Verlag.

42. Stützle, T. (1998). An ant approach to the flow shop problem. In *Proceedings of the Sixth European Congress on Intelligent Techniques & Soft Computing (EUFIT '98)*, *Aachen, Germany* (pp. 1560–1564). Berlin: Verlag Mainz.

43. Stützle, T., & Dorigo, M. (1999). ACO algorithms for the quadratic assignment problem. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 33–50). London: McGraw-Hill.

44. Zlochin, M., & Dorigo, M. (2002). Model-based search for combinatorial optimization: A comparative study. In *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN2002)* (pp. 651–662).