ATHABASCA UNIVERSITY

AUTOMATIC IDENTIFICATION OF LEARNING STYLES
AND WORKING MEMORY CAPACITY FROM STUDENT BEHAVIORS
USING COMPUTATIONAL INTELLIGENCE ALGORITHMS

BY

JASON BERNARD

A thesis submitted to the Faculty of Graduate Studies
In partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE in INFORMATION SYSTEMS

ATHABASCA, ALBERTA
November, 2016

Approval of Thesis

The undersigned certify that they have read the thesis entitled

“Automatic Identification of Learning Styles and Working Memory Capacity from Student Behaviors Using Computational Intelligence Algorithms”

Submitted by

Charles Jason Bernard

In partial fulfillment of the requirements for the degree of
Master of Science in Information Systems

The thesis examination committee certifies that the thesis
and the oral examination is approved

Internal Co-Supervisor:
Dr. Sabine Graf
School of Computing and Information Systems
Athabasca University

External Co-Supervisor:
Dr. Elvira Popescu
Faculty of Automation, Computers and Electronics
University of Craiova

Committee Member:
Dr. Ting-Wen Chang
Smart Learning Institute
Beijing Normal University

External Examiner:
Dr. Carolina Corredor
School of Studies in Virtual Environments
EAN University

December 7, 2016

Dedication

This work is dedicated to Ms. Sorel, my grade 1 & 2 school teacher. You more than anyone made me fall in love with learning and because of that this thesis exists.

Acknowledgments

I would like to thank all of three of my supervisors for the mentoring they've provided me during this research. Dr. Sabine Graf has provided me with much advice on how to properly do research, how to make a valid argument and on scientific writing. Although I recognize that I have areas for improvement, I know that I've learnt much from her teachings. I honestly could not have asked for a better mentor, and if I ever become anything of a researcher it will certainly be due to her advice. Dr. Ting-Wen Chang has also provided great advice but helped by constantly reminding me that I'm at the start of my research career and that it takes time and patience to acquire real mastery. Dr. Elvira Popescu also provided a lot of advice particularly with respect to writing. She also told me quite frequently not to be too anxious which is very helpful as I think graduate students are natural worriers.

I also would like to thank the following organizations for their support: NSERC, Athabasca University, Alberta Innovates – Technology Future (AITF), Alberta Innovation and Advanced Education (AIAE) and the Prof. Ram Kumar Memorial Foundation. NSERC supports Dr. Graf's research endeavors and so that, in turn, makes this research possible. The AITF and AIAE helped me by allowing me to focus on my studies and not on day-to-day finances. The AITF, AIAE, Athabasca University and the Prof. Ram Kumar Memorial Foundation helped finance the dissemination of the results of this research

Lastly, I would like to thank my friends and family for their support during this research. They always believed that I could and would successfully finish this thesis. I appreciate my mother asking me questions about my research, which helped me understand it better myself.

Abstract

By identifying students' learning styles and working memory capacity (WMC) personalized scaffolding techniques can be used, either by teachers or adaptive systems to offer students individual recommendations of learning activities. Such personalization has been shown to have a positive effect on learning outcomes. Traditionally, learning styles and WMC have been identified by dedicated test. However, these tests have certain drawbacks (e.g., students have to spend additional time on them, etc.). Therefore, recent research aims at automatically identifying learning styles and WMC from students' behavior in learning systems. This thesis presents an investigation into using different computational intelligence algorithms to build automatic approaches to more precisely identify learning styles and WMC. An evaluation of these approaches using real student data shows that most improve precision over existing leading approaches. However the best result for learning styles was a hybrid architecture improving precision styles to 80.4% and an evolving artificial neural network improving precision for WMC to 88.0%. By increasing the precision of learning styles and WMC identification, more accurate interventions can be made to better support students while learning.

Table of Contents

List of Tables

List of Figures and Illustrations

List of Symbols, Nomenclature, and Abbreviations

ACC	Accuracy
ACS	Ant Colony System
ACO	Ant Colony Optimization
AI	Artificial Intelligence
ANN	Artificial Neural Network
BN	Bayesian Network
CI	Computational Intelligence
DeLeS	Detecting Learning Styles
DeWMC	Detecting Working Memory Capacity
EANN	Evolving Artificial Neural Network
EANN/R	Recurrent Evolving Artificial Neural Network
FSLSM	Felder-Silverman Learning Styles Model
GA	Genetic Algorithm
HMM	Hidden Markov Model
ILS	Index of Learning Styles
LACC	Lowest Accuracy
LMS	Learning Management System
LO	Learning Object
LSID	Learning Style Identifier
MLP	Multilayer Perceptron
PSO	Particle Swarm Optimization
SIM	Similarity
SISO	Simplify and Solve
WMC	Working Memory Capacity

WMCID	Working Memory Capacity Identifier
%Match	Percentage of Students Matched

Chapter I - Introduction

Being able to identify a student's learning styles and working memory capacity (WMC) allows a student to learn better and faster through several means. By understanding themselves, students are able to capitalize on their own strengths and make better decisions to support self-regulated learning. Furthermore, learning systems may be personalized to the students by providing them advice, recommendations or material adapted to their abilities and preferences. Personalizing content to the learning styles and WMC of students has been found to be beneficial to learning in several ways such as improving motivation (Cordova & Lepper, 1996; Popescu, 2010), learning outcomes (Bajraktarevic, Hall, & Fullick, 2003; Paas, Renkl, & Sweller, 2004), learning transfer (Moreno, 2004; Van Merriënboer, Schuurman, De Croock, & Paas, 2002) and reducing the time needed to learn (Cooper, 1998; Graf, Chung, Liu, & Kinshuk, 2009).

As a first step towards personalization, the learning styles and WMC of students must be identified. Classically, this is done using dedicated tests such as the Index of Learning Styles (ILS) (Felder & Solomon, 1998) and (Operation Span Task) OSPAN (Turner & Engle, 1989), which although valid and reliable (Felder & Spurlin, 2005; Klein & Fiss, 1999) have at least two notable drawbacks. The main drawback is that they take the student away from the learning task by requiring time and effort to complete the test. Furthermore, these tests can misidentify due to fatigue (Gohar et al., 2009), stress (Beilock & Carr, 2005), misconceptions or lack of effort from the student. To overcome these drawbacks, automatic approaches have been proposed building a student model from the students' behaviors when using a learning management system (LMS) (Chang, El-Bishouty, Graf, & Kinshuk, 2013; García, Amandi, Schiaffino, & Campo, 2007; Graf, Kinshuk, & Liu, 2009).

Such automatic approaches currently peak at a precision from 70.9% to 79.9% for learning styles (Graf, Kinshuk et al., 2009) (a range due to learning styles having multiple dimensions) and 80.9% for WMC (Chang, El-Bishouty, Kinshuk, & Graf, 2016), so there is room for improvement. This research aims to improve the precision of automatically identifying learning styles and WMC. Whereas most research on automated approaches identify their own student modeling process, this research capitalizes on the work already done by leading automated approaches in two fashions. First, computational intelligence (CI) algorithms are used to optimally weight the rules of the leading automated approaches; thereby improving precision. Second, since the behavior patterns identified by these leading automated approaches are known to provide fairly precise results, these same behavior patterns are used as inputs into different CI algorithms towards producing a new approach which is more precise. This is done in two phases described as follows.

In the first phase, four approaches are developed and evaluated to improve the precision of identification each using a separate CI algorithm: artificial neural network (ANN), ant colony system (ACS), genetic algorithm (GA) and particle swarm optimization (PSO). For learning styles identification these approaches are called LSID-ANN, LSID-ACS, LSID-GA and LSID-PSO (with LSID meaning Learning Style Identifier) and similarly for WMC the approaches are called WMCID-ANN, WMCID-ACS, WMCID-GA and WMCID-PSO (with WMCID meaning Working Memory Capacity Identifier). For the second phase, the results from the first phase are analyzed and a hybrid CI algorithm is selected to overcome any weakness discovered. For learning styles, a loosely coupled hybrid architecture was selected and an architecture called

"Simplify and Solve" (SISO) was developed to build the approach LSID-SISO. For WMC, an evolving artificial network (EANN) is selected, including the recurrent topology variant (EANN/R) and used to build WMCID-EANN and WMCID-EANN/R. WMCID-SISO is also built to show that the SISO architecture works specifically due to observations made on the identification of learning styles (i.e. WMCID-SISO was not expected to improve results for WMC).

The remainder of this thesis is structured as follows. Chapter 2 provides a review of literature through several sub-sections on: student modeling, learning styles models, WMC, a survey of existing automatic approaches for identifying learning styles and WMC and a background on algorithms used in this research. Chapter 3 describes all of the LSID and WMCID approaches and how the CI algorithms were adapted to identify learning styles and WMC. Chapter 4 discusses the methods used in this thesis, including the data, performance metrics and describes each of the LSID and WMCID approaches. Chapter 5 provides the results from each of the algorithms for identifying learning styles and WMC. The LSID and WMCID approaches are compared to other approaches and each other to identify the best approaches. Chapter 6 analyzes the results from the LSID and WMCID approaches and the execution of the algorithms. Chapter 7 concludes the thesis and discusses the future direction of the research.

Chapter II - Review of Literature

This chapter examines the existing research which forms the underlying basis of this study. Student modeling is explored by looking at the information provided by student models, how such models are built and how they can be used to support learning. This is followed by a discussion on learning styles and WMC. Afterwards a survey of existing automatic approaches for identifying learning styles and WMC is presented. This chapter concludes with a discussion on the CI algorithms used to build the identification approaches in this research.

2.1 Student Modeling

A student model is a representation of a student that provides predictions about the characteristics of the student (VanLehn, 1988). Student modeling is the process of “creating a student model” (Self, 1994) or “the process of building and updating the student model” (Graf, 2007). Student modeling has origins primarily from research on intelligent tutoring systems (ITSs) (Kass, 1989; Sison & Shimura, 1998; VanLehn, 1988); however, more recently student modeling has been used to implement educational adaptive multimedia systems (Brusilovsky & Millán, 2007; Chrysafiadi & Virvou, 2013; Encarnação, 1997; Jia, Zhong, Wang, & Yang, 2009) which seek to provide an optimal learning environment for students by adapting to their individual characteristics. Student models are built by transforming known information about a student into a prediction about their characteristics. The heart of this thesis research is about student modeling as this research seeks to transform student’s behaviors into a prediction of their learning styles or WMC. The remainder of this section examines three questions:

1. What information do student models provide?
2. How are student models constructed?
3. How may the student model be used to support learning?

2.1.1 Information Provided by Student Models

This sub-section will examine the information typically found in student models and some of the ways in which the information is represented. The information provided by student models may be broken down broadly into the following five categories: “knowledge, interests, goals, background and individual traits” (Brusilovsky & Millán, 2007). Each of the categories are described as follows from the work of Brusilovsky and Millán (2007).

The knowledge category represents what the student knows about a particular subject. To model a student’s knowledge level a number of models have been proposed. The simplest is the scalar model, which uses either a quantitative (e.g. 0 to 10) or qualitative (e.g., none, poor, average, good, excellent) scale to describe the user’s knowledge. Despite the simplicity, such scalar models have been successfully used to implement educational adaptive learning systems (Beaumont, 1994; Boyle & Encarnacion, 1998; Encarnação, 1997). Although scalar models can describe a level of expertise, they do not describe what information within the topic the student knows. To address this, VanLehn (VanLehn, 1988) proposed the overlay model which breaks down a topic into fragments where each fragment describes a particular expert level piece of knowledge. The simplest overlay model describes whether the student knows any particular fragment with a true / false value. The drawback to this type of overlay is it does not clearly convey a level of expertise. Thus, the scalar model and overlay model may be combined such that the student's knowledge is described as a scalar value for each fragment, and the overall expertise level for the topic as a weighted average of the individual fragments (Brusilovsky & Millán, 2007).

Knowledge modelling may be further extended to describe the missing or insufficient expertise due to misconceptions, or perturbations (Kass, 1989; VanLehn, 1988). A list of possible perturbations can be created by examining literature or through analysis of students' behaviors to find common errors (VanLehn, 1988). The difficulty is in assuring a relatively complete list of possible perturbations, and what to do when a student displays a perturbation that is not on the list. Describing all possible perturbations from literature is very costly, in terms of effort, and ongoing as it can never be complete with certainty. Extracting perturbations from students' behaviors is currently a manual process, no literature could be found to do this in an automatic fashion, and so is also costly in terms of effort. Additionally, it is subject to human bias as the human expert must decide if a particular error is a perturbation, if so is it an existing perturbation and if not how should it be objectively described. For example, if the perturbation is described as a logical rule, it must be crafted very carefully so that it will not encompass other perturbations or worse desirable behavior. These issues should be regarded as open research questions for the knowledge category. Regardless, overall, the student's knowledge state for a given topic may be described as the overlay model plus the set of perturbations.

The interest category pertains to the areas of interest for the student, both academic and non-academic. For example, it might describe that a student has an interest in sports and mathematics. The interest category may be described either as concepts or by keywords. Interests are generally thought to be an element of student models (Brusilovsky & Millán, 2007; Surjono & Maltby, 2003; Walkington, 2013) perhaps because they play a prominent role in user models for adaptive hypermedia systems in other domains (e.g. sales and marketing).

The goal category is relatively straightforward as it simply describes what it is that the student wishes to learn. Providing adaptive material based on goals has been well-researched (Brusilovsky, 1992; McArthur, Stasz, Hotta, Peter, & Burdorf, 1988; McCalla, Bunt, & Harms, 1986; Ueno, 2005) although no example in literature could be found for adaptiveness based only on students' goals. Rather in literature the student model combined knowledge level with students' goals. In the literature, the goals in the system are defined manually from expert knowledge.

The background category describes the student's previous experience. This may include items such as profession, previous work or study experience, native and non-native languages. Background information is difficult to capture by monitoring and so is generally explicitly provided by the student and is modelled simply as a stereotype (Brusilovsky & Millán, 2007).

The individual traits category contains those characteristics which describe a person as an individual. These characteristics generally pertain to how the person thinks and feels such as "personality traits (e.g. introvert/extrovert), cognitive styles (holist/serialist), cognitive factors (e.g., working memory capacity) and learning styles" (Brusilovsky & Millán, 2007). Although such traits may be captured by specialized psychological tests there are a growing number of automatic approaches to capture these characteristics by monitoring the student, such as this research does for learning styles and WMC.

2.1.2 Constructing the Student Model

There are two approaches towards gathering data for a student model: collaborative and automatic (Brusilovsky, 1996). In the collaborative approach, the student provides the information, typically by simply being asked. Sources for such data might include the student's personnel file (gender, age, etc.), academic record (courses completed, grades), questionnaires (such as the ILS (Felder & Solomon, 1998)) and standalone questions ("Did you find this learning object helpful?" or "Please select a learning goal for this session"). The main drawback to ask the student is that for some questions the student may answer as in accordance to expectations, internal or external, instead of truthfully. For example, a student may be unwilling to say that they found a learning object unhelpful for fear of angering their professor. Alternatively, on the ILS (Felder & Solomon, 1998) for example, students may answer how they wish, how they acted or how they think a good student is supposed to act instead of how they actually feel. A proposed method to lessen this effect is the "Do It Yourself" method (Bull, 1998) which asks the student to quantify how certain they are about the response. Although this is not proof against a student answering in accordance with expectations, it encourages students to self-evaluate and reflect on their answers; thereby, improving the quality of responses.

With the automatic approach data is gathered from the student without direct interaction, although this should not imply that is gathered unethically, i.e. without their knowledge or consent. Typically, the data is gathered from students' interactions with the system which includes items such as: content preferences, navigational behavior, biometrics, keystrokes, mouse clicks, forum posts or conversation logs (textual or verbal). From these sources useful information is inferred about the student for the model and much research has been done on how to transform this data. This list is certainly not exhaustive; however, some examples include identifying learning styles (García et al., 2007; Graf, Kinshuk et al., 2009; Latham, Crockett, McLean, & Edmonds, 2012), cognitive styles (Frias-Martinez, Chen, & Liu, 2007), WMC (Chang et al., 2013), reading skill (Beck & Chang, 2007), or students at-risk of poor performance in a course (Jayaprakash, Moody, Lauría, Regan, & Baron, 2014; Yu, Own, & Lin, 2001).

Automatic approaches used to build student models are either data-driven or literature-based (Graf, 2007). Under the data-driven paradigm, student data, for example behavior data, is mined to produce the model rules or an AI/CI algorithm (e.g. Bayesian, decision tree learning) may be trained on the data to produce the model. The drawback to the data-driven paradigm is the requirement for data of both sufficient quantity and quality to develop a precise student model. The literature-based approach builds the student model by developing rules based on expert knowledge (or literature) (Graf, 2007). The drawback to this approach is that the rules which can be extracted from literature may be either too simple or difficult to extract and encode logically. As previously mentioned, this research aims to improve upon existing two leading literature-based approaches (Chang et al., 2013; Graf, Kinshuk et al., 2009). This is done in three fashions. The first uses behavior data to train an ANN, while the second uses optimization algorithms to find optimal weights for the rules produced by the literature-based approaches selected (Chang et al., 2013; Graf, Kinshuk et al., 2009). The third fashion selects the best algorithm from the first two and then uses hybrid CI algorithms to address any weaknesses.

2.1.3 Supporting Learning with Student Models

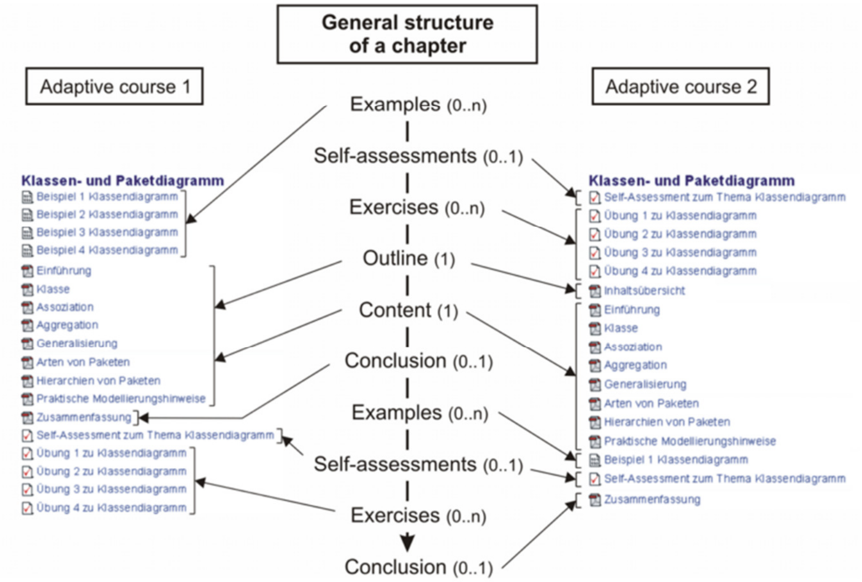
The ultimate goal of using a student model is to know something about the students which can then be used to allow students to learn faster or better. This is generally accomplished in one of three ways. First, the student will be better informed of their strengths and weakness and can make better choices for self-regulated learning with effective self-regulated learning being a predictor of higher performance (Pintrich & De Groot, 1990). Second, teachers are supported by understanding how their students learn or where they struggle and thus are more able to provide them with appropriate interventions (Delozanne, Grugeon, Previt, & Jacoboni, 2003; Graf, Kinshuk et al., 2009; Lin, 2004). Third, for online and blended learning, the learning environment for the student may be personalized to allow them to learn faster or better (Graf, 2007; Klašnja-Milićević, Vesin, Ivanović, & Budimac, 2011; Popescu, 2010).

Although personalization may be provided in a large variety of ways, three of the most common are: "adaptive content selection, adaptive navigation support, and adaptive presentation" which are described as follows from Brusilovsky's work (2012).

Adaptive content selection is used when the student searches for information. Rather than returning the data in a non-adaptive manner (e.g., in alphabetical order or in order of being found) the system sorts the data so that the information which would be most helpful or appealing to the individual student appears higher in the list. Various studies have examined successful ways to implement adaptive content selection (Brajnik, Guida, & Tasso, 1987; Brusilovsky, 1992; Chen & Kuo, 2000). For example, Chen and Kuo (2000) use query feedback from the student to dynamically model what the student believes a term means, which will change over time as the student gains more knowledge and experience. They then use the intended meaning of search terms to find the most relevant search items. One example they provide is that for a particular student the search term "watermark" may implicitly represent "information hiding". Adaptive content selection aims for three interrelated outcomes: make the system easier to use (by making it simpler for the student to find the information they seek), increase system adoption by students, and increase student satisfaction (Mulwa, Lawless, Sharp, Arnedillo-Sanchez, & Wade, 2010).

Adaptive navigation support is provided to the student by ordering, hiding or recommending links to particular items in accordance with the student's characteristics (Graf, 2007; Mampadi, Chen, Ghinea, & Chen, 2011). For example, Graf et al. (2009) investigated the effects of adaptive navigational support on three of the four FSLSM dimensions with the V/V dimension excluded. The experiment divided the students into three groups with adaptivity provided by altering the placement and/or number of links to learning objects (LOs) with each chapter. For the first group (matched), the course was adapted to match their learning styles while for the second group (mismatched) the course was adapted to not match their learning styles. The third group (standard) received a non-adapted version of each chapter. For example, Figure 1 shows that for "adaptive course 1" there are 4 links to examples and they are placed at the top of the list, while for "adaptive course 2" there is only 1 link to an example and it is placed nearer the bottom. The study found that adaptivity is beneficial and works differently for students with different learning styles. When provided with mismatched material, students with active or sequential preferences were found to spend more time with the material, thus learning inefficiently. They found that reflective, sensing and global students seemed to recognize the unsuitability of mismatched material and requested additional material more frequently.

Figure 1. Chapter outlines adapted to student learning styles (Graf, 2007)


General structure of a chapter
Adaptive course 1
Adaptive course 2
Examples (0..n)
Self-assessments (0..1)
Exercises (0..n)
Outline (1)
Content (1)
Conclusion (0..1)
Examples (0..n)
Self-assessments (0..1)
Exercises (0..n)
Conclusion (0..1)
Klassen- und Paketdiagramm
Beispiel 1 Klassendiagramm
Beispiel 2 Klassendiagramm
Beispiel 3 Klassendiagramm
Beispiel 4 Klassendiagramm
Einführung
Klasse
Assoziation
Aggregation
Generalisierung
Arten von Paketen
Hierarchien von Paketen
Praktische Modellierungshinweise
Zusammenfassung
Self-Assessment zum Thema Klassendiagramm
Übung 1 zu Klassendiagramm
Übung 2 zu Klassendiagramm
Übung 3 zu Klassendiagramm
Übung 4 zu Klassendiagramm
Klassen- und Paketdiagramm
Self-Assessment zum Thema Klassendiagramm
Übung 1 zu Klassendiagramm
Übung 2 zu Klassendiagramm
Übung 3 zu Klassendiagramm
Übung 4 zu Klassendiagramm
Inhaltsübersicht
Einführung
Klasse
Assoziation
Aggregation
Generalisierung
Arten von Paketen
Hierarchien von Paketen
Praktische Modellierungshinweise
Beispiel 1 Klassendiagramm
Self-Assessment zum Thema Klassendiagramm
Zusammenfassung

Adaptive presentation modifies the contents of LOs to match the characteristics of the student. Two common research techniques are hiding, showing or ordering content within the LO (Carver, Howard, & Lane, 1999; Melis et al., 2001; Popescu, 2010) and providing adaptive scaffolding, which are hints or guidance tailored for the student based on the problems they are having or their characteristics (Azevedo, Cromley, Winters, Moos, & Greene, 2005; Ley, Kump, & Gerdenitsch, 2010; Segedy, Biswas, Blackstock, & Jenkins, 2013). For example, Popescu (2010) developed the tool WELSA to adapt course web pages based on students' learning styles. The student model for WELSA is built from mining a log of the students' activities. One way WELSA adapts web pages is by initially showing or hiding content, for example a student with a visual preference will have images shown and text hidden and inversely for a student with a verbal preference. Additionally, the student is provided with a recommendation cue to the learning objects best suited to their preferences. Students are free to view the hidden content by expanding the corresponding LO. Popescu evaluated student motivation and satisfaction when provided with matched and mismatched content by use of a survey. It was found that both satisfaction and motivation are improved with the matched content.

2.2 Learning Styles

This section discusses the Felder-Silverman Learning Styles Model (FSLSM), starting with an overview of the learning styles as a whole. Then the FSLSM itself is discussed with a look at the psychological questionnaire, the Index of Learning Styles (ILS) (Felder & Solomon, 1998), used classically to identify learning styles under this model. This is followed by an examination of each of the FSLSM dimensions, including proposals on possible curriculum modifications for each learning style dimension. Lastly, the validity and reliability of the ILS is discussed.

There exist a variety of definitions for learning styles, such as "strengths and preferences in the ways they take in and process information" (Felder & Soloman, 2000), "a description of the attitudes and behaviours which determine an individual's preferred way of learning" (Honey & Mumford, 1992), "characteristic cognitive, affective and psychological behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment" (Keefe, 1979). One common element among these definitions is the idea that each has their own individual preference towards how they may best learn. Understanding learning styles unlocks one possible explanation for why some students struggle and in turn to then help them learn better and faster. Felder & Silverman (1988) states that prior to research into learning styles the student would typically be blamed for a failure to learn, and although this might be warranted sometimes, in other cases the fault lies with the curriculum not reaching some students. They argue that traditional curricula only appeals to particular learning styles and students which did not share these learning styles may suffer and not by a lack of ability or effort.

To describe these preferences, several models have been produced such as those by Felder-Silverman (1988), Kolb (1971), Pask (1976) and Honey and Mumford (1992). Models, such as the four listed, describe the preferences as labels and/or dimensions, sometimes in opposition to each other. For example, Kolb's model has two opposing dimensions, abstract/concrete and active/reflective from which four learning styles are defined as labels: converging (abstract, active), diverging (concrete, reflective), assimilating (abstract, reflective) and accommodating (concrete, active).

Being able to understand students' learning styles allows both teachers and students to be supported for the benefit of the student. Teachers are supported by giving them an initial point of understanding and so allowing them to make more appropriate interventions for a student struggles (Graf, Kinshuk et al., 2009). When a student understands themselves this is not only empowering for the student (Felder & Spurlin, 2005), but allows them to make better choices for self-regulated learning which can help them achieve better performance (Pintrich & De Groot, 1990). Finally, as discussed in the section on student modeling there is much research on using adaptive technologies to personalize the learning environment towards the students' learning preferences (Bajraktarevic et al., 2003; Brajnik et al., 1987; Brusilovsky, 1992; Chen & Kuo, 2000; Graf, 2007; Klašnja-Milićević et al., 2011; Mampadi et al., 2011; Mulwa et al., 2010; Popescu, 2010) with benefits such as an improvement in satisfaction (Popescu, 2010), learning outcomes (Bajraktarevic et al., 2003) or a decrease in the time needed to learn (Graf, Chung et al., 2009).

Although there is much appeal to learning styles, there does exist some criticism in literature. With respect to learning styles in general, Coffield et al. (2004) state that due to the intrinsic appeal of learning styles, it has become commercialized with exaggerated claims of efficacy. They state that learning styles research encompasses a very large body of work with occasionally divergent viewpoints while it is oft treated as "united in its thinking" (2004). The effect, in particular from industry, is that advice provided to practitioners has been summarized from multiple, often contested, works and results in being simplified to the point of not being very useful. They also argue that in some cases claims made from researchers without the support of empirical data are used to provide advice to practitioners. This research has taken such criticism into account and focused the literature review on works which avoid exaggerated claims and mainly focuses on those works which have been evaluated using real student data.

Both studies of Coffield et al. (Coffield et al., 2004) and Pasher et al. (Pashler, McDaniel, Rohrer, & Bjork, 2008) criticize the practicality of capitalizing on learning styles in the face-to-face context. In a classroom with many students and so many different learning styles represented there is no possibility that a teacher could adapt to each learning style. Solving that issue by splitting classrooms by learning styles is also logistically impractical in large scales (Coffield et al., 2004). However, proponents for learning styles do not generally argue that in the face-to-face context the teacher must adapt their teaching style to each student or that the classroom should be separated by learning style. Rather, it is argued that since traditional curriculum satisfies only a narrow range of learning styles, a curriculum which considers different learning styles should be adopted to meet the needs of more students (Felder & Spurlin, 2005). In any case, the argument that the effort to provide individual adaptation is too high has decreasing validity with research showing that adaptive learning systems are quite capable of automatically adapting to any number of students (Brajnik et al., 1987; Brusilovsky, 1992; Chen & Kuo, 2000; Graf, 2007; Klašnja-Milićević et al., 2011; Popescu, 2010) and with the increase in the use and desirability of adaptivity in LMSs (Dahlstrom, Brooks, & Bichsel, 2014).

With respect to the benefits of learning styles, especially with respect to content matching, there is some criticism that the effects tend to be small or contradictory (Coffield et al., 2004; Pashler et al., 2008). From this, they do not conclude that matching content to learning styles is invalid but rather that the research on how to successfully match content with learning styles is incomplete and that more research is required. More specifically, they suggest that research should examine how learning styles should be intermixed with other characteristics to be successful at benefiting students (Coffield et al., 2004). Although more research may still be needed, studies examining intermixing characteristics have been done. Two independent studies (Limongelli, Sciarrone, Temperini, & Vaste, 2009; Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2003), evaluated intermixing knowledge level with learning styles and found a positive effect on learning outcomes. Also, despite criticism on the benefits of learning styles, Coffield et al. (2004) agree that identification of learning styles is useful as a means of self-awareness for students.

One of the major challenges with respect to learning styles pertains to how they are identified and this challenge is directly addressed by this research. Classically, learning styles are identified through the use a psychological questionnaire. For example, the Kolb model uses the Learning Styles Inventory (Kolb & Hay, 1999), the Honey and Mumford model uses the Learning Styles Questionnaire (Honey & Mumford, 2006) and the Felder-Silverman learning styles model (FSLSM) uses the Index of Learning Styles (ILS) (Felder & Solomon, 1998). Although the questionnaires for Kolb's model and the FSLSM are considered valid and reliable (Felder & Spurlin, 2005; Willcoxson & Prosser, 1996; Wilson, 1986) they do have some notable drawbacks. First, it is intrusive to the learning task as students must fill in the questionnaire in addition to learning activities. Second, as previously discussed, a questionnaire may be influenced by other factors than just a student's learning styles. A student's perceived importance of the questionnaire can lead to a misidentification of their learning styles as they may answer the questions very quickly without much thought. Further, student's answers may be biased by personal misconceptions or from perceived expectations. To overcome these drawbacks, automatic approaches, such as those presented in this thesis, have been researched to identify students' learning styles from their behavior (Carmona, Castillo, & Millán, 2008; Cha et al., 2006; Dorça, Lima, Fernandes, & Lopes, 2013; García et al., 2007; Graf, Kinshuk et al., 2009; Latham et al., 2012; Özpolat & Akar, 2009; Villaverde, Godoy, & Amandi, 2006). An automatic approach reduces intrusiveness by working in the background as the student uses the learning system. Automatic approaches are not influenced by student's perceived importance, preconceptions or expectations with respect to learning styles as only their actual behaviors are considered.

2.2.1 Felder-Silverman Learning Style Model

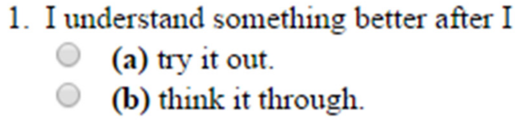
The FSLSM (Felder & Silverman, 1988) was proposed in 1988 with the aim of providing insight to faculty on how to provide a better learning environment for students who were not being reached by existing curricula. Felder and Silverman thought that low performance for engineering students was usually blamed on the student, when it might be better explained that the standard curricula was not suited to their learning styles. Felder and Silverman propose not only a learning styles model, but discuss how to best accommodate different learning styles in the classroom (their work predates the rise in blended and online learning).

Originally consisting of five dimensions, the most recent version of the FSLSM consists of four dimensions: active/reflective (A/R), sensing/intuitive (S/I), visual/verbal (V/V) and sequential/global (S/G). The V/V dimension was originally called visual/auditory; however, this was changed as it was unclear where a preference for reading should fall. Some thought that it should be in the visual preference as it generally requires the use of sight. By changing auditory to verbal, it makes it clearer that the opposite preference to visual is linguistics whether read or heard. The fifth dimension in the FSLSM was inductive / deductive and was removed even though it does describe a learning preference. Felder states in the author's preface (Felder & Silverman, 1988) that although an inductive curriculum is more effective, most students tend towards a deductive preference. He feared that faculty would use the deductive preference as justification to continue with the traditional, but less effective, deductive-based curricula.

For this research, the Felder-Silverman learning styles model (FSLSM) (Felder & Silverman, 1988) has been selected for several reasons. To begin, the FSLSM brings together different elements from the models by Kolb (1971), Pask (1976) and the Myers-Briggs personality inventory (Myers-Briggs, 1962). The FSLSM uses four dimensions, described in subsequent sub-sections, active / reflective (A/R), sensing / intuitive (S/I), visual / verbal (V/V) and sequential / global (S/G) allowing the student's learning styles to be described in great detail. Where other models tend to use labels the FSLSM allows each dimension to vary from +11 to -11, in increments of 2. This more accurately describes learning styles as a tendency as opposed to an absolute behavior allowing students' learning styles to be described more deeply than with labels. Research has found that the FSLSM is well-suited to use as a model for providing personalization (Kuljis & Liu, 2005) and is used commonly in literature (Bajraktarevic et al., 2003; Cha et al., 2006; García et al., 2007; Graf, 2007; Limongelli et al., 2009; Villaverde et al., 2006). Lastly, there exists a valid and reliable questionnaire, the Index of Learning Styles (ILS) (Felder & Solomon, 1998), for identifying Felder-Silverman learning styles.

To identify a student's learning styles under the FSLSM, the Index of Learning Styles (ILS) (Felder & Solomon, 1998) is used. It consists of 44 questions each with two choices where the responses are associated to both poles of a learning style dimension (a sample of question #1 is shown in Figure 2). In the example, the response "a" denotes an active preference and “b” indicates a reflective preference. The relationships between the questions and preferences are shown in Table 1. A response for the active, sensing, visual and sequential preferences are assigned a value of +1 and -1 is assigned to the other response. As each dimension has 11 questions assigned to it, each dimension in the FSLSM is described as a scale from -11 to +11 in increments of 2 (switching responses creates a change of 2 points as it shifts from +1 to -1 or vice versa). Thus, the higher values indicate a strong active, sensing, visual or sequential preference and inversely lower values a reflective, intuitive, verbal or global preference. The relative value should not be interpreted to imply that one preference is better than another, i.e. an active preference (+5 to +11) is not better than a reflective preference (-5 to -11) or a balanced preference (-3 to +3). Since a scale is used the FSLSM describes preferences as a tendency instead of an absolute behavior, and is a differentiating feature of the FSLSM from other models which tend to use labels only.

Figure 2. Sample question from the ILS (Felder & Solomon, 1998)



1. I understand something better after I
   ○ **(a)** try it out.
   ○ **(b)** think it through.

Table 1. Relationship between learning style dimension and Index of Learning Style questions

Dimension	Question #
Active/Reflective	1,5,9,13,17,21,25,29,33,37,41
Sensing/Intuitive	2,6,10,14,18,22,26,30,34,38,42
Visual/Verbal	3,7,11,15,19,23,27,31,35,39,43
Sequential/Global	4,8,12,16,20,24,28,32,36,40,44

The remainder of this section will focus on the latest version of the FSLSM (Felder & Soloman, 2000) starting with describing the four dimensions and their origins from literature. Also, discussed are Felder & Silverman's (1988) recommendations for learning material and activities. The recommendations have a bias towards science education as the FSLSM was originally developed for engineering faculty in a classroom setting. However, since then the FSLSM has been shown to be very appropriate for use in eLearning (Carver et al., 1999; Kuljis & Liu, 2005). Lastly, the validity and reliability of the ILS (Felder & Solomon, 1998) will also be discussed in the final sub-section.

2.2.1.1 Active / Reflective Dimension

The A/R dimension is derived mainly from Kolb's learning styles model (1971) which consists of two dimensions: processing and perception. The processing dimension is described by two labels, active experimentation (doing) and reflective observation (watching). The active / reflective dimension is also loosely inspired by the extrovert / introvert attitude types from Jung's personality types (1971), later used in the Myers-Briggs Type Indicator (MBTI) (1962), which describe a person as focussed on the external world or internal thought respectively.

Both the Kolb learning styles model and the FSLSM state that the A/R dimension describes how a student converts perceived information into knowledge. Thus, for students with the active preference, this means doing something with the information, while reflective students prefer the opportunity to consider the information internally. In context of curriculum, Felder & Silverman (1988) emphasize that neither of these students prefer to be passive, and so lecturing alone is not effective for either preference. For active students, discussions and experimentation are good ways to help them learn material. For reflective learners within the classroom discussions and brainstorming sessions may also be effective; however, they should be done in smaller groups since reflective learners tend to be more introverted and might not participate in discussions involving a large classroom. With respect to lecturing or content in a learning system, as active students prefer to experiment, such students prefer material which presents practical means to solve problems (practical problem solving is also related to the sensing learning style described below) to support their experimentation (Felder & Silverman, 1988). While reflective students will prefer more theoretical material since this promotes inner thought and understanding.

2.2.1.2 Sensing / Intuitive Dimension

The S/I dimension is derived from Jung's personality types (1971) which posited that people could be categorized by two functions: perceiving and judging. The perceiving function is, in turn, sub-divided into two categories, sensation and intuition. The perceiving function is described as relating to how information is gathered. Thus, with the S/I dimension, as with Jung's personality types, the sensing preference means that such students gather information by the use of their senses, i.e. interacting with the real world. Students with an intuition preference gather information indirectly, through the use of speculation or imagination.

Sensing students prefer facts, experimentation and are patient with details. Intuitive students prefer principles, theory and become bored with details or repetition. In general, sensing students may process verbal material slower than intuitive students possibly because sensing students are less comfortable with symbols, which includes words. Intuitive students tend to work more quickly; however, this tends to make them careless as they may not pay attention to details.

In terms of curricula, learning material should be split so that there is a balance of facts and practical problem solving versus theory and principles. For scientific courses, when teaching a theory providing examples of predictions made by theory is helpful for sensing students. This can be followed by a description of how the theory is developed, which will be helpful for intuitive students. For learning activities, sensing students will prefer to drill exercises and conduct experiments, while intuitive students prefer opportunities to use logic to develop their own theory.

2.2.1.3 Visual / Verbal Dimension

The V/V dimension describes how students prefer to receive information. Barbe et al. (1979) developed the VAK model which describes how people receive information using three learning modalities: visual, auditory and kinesthetic. The visual modality relates to seeing material, the auditory modality with hearing information (and includes reading) and the kinesthetic modality with feeling or tasting material. The kinesthetic modality is not very applicable to higher education where most activities involving lectures (N.B. it would also be impractical for online learning) and so was not included in the FSLSM. The VAK model would later be expanded to the VARK model which included a reading/writing modality (Fleming, 1995) in order to differentiate the spoken word from the written word. As previously discussed, for the FSLSM the auditory preference was renamed verbal as it included a preference for reading from the outset.

Most students exhibit a preference for the visual modality (Barbe & Milone Jr, 1981; Felder & Spurlin, 2005); however, most classrooms are a verbal experience, since they rely heavily on lecturing. The recommendation by Felder & Silverman is straightforward, that to appeal to visual students graphs, charts, diagrams and flow charts should be presented alongside of the lecture (or as learning material in a learning system). Additionally, live demonstrations (or video) of processes are an effective means to reach visual students. Since most learning material is already either auditory (lectures) or involves reading nothing extra is needed for verbal students. Lastly, learning is reinforced when all modalities (including kinesthetic) are used in concert with each other regardless of the student's visual or verbal preference (Barbe et al., 1979; Dunn, DeBello, Brennan, Krimsky, & Murrain, 1981). Thus, verbal students who are already well serviced by the existing curricula, lectures and reading material, will benefit from visual material and kinesthetic activities as well.

2.2.1.4 Sequential / Global Dimension

Conversation Theory (Pask, 1976) proposes that learning between cognitive systems (much of Pask's work was in cybernetics and so was not exclusive to human interaction) occurs by conversation about the material. Pask proposed that to make learning easier material should be organized appropriately; however, that there exists two organization strategies: serialist and holist. The serialist strategy is to organize material linearly, while the holist strategy organizes material as a conceptual framework (a non-linear series of topics within the subject matter). The S/G dimension describes the preference a student has for how information is organized. The sequential student prefers the serialist organization strategy, while the global student prefers the holist organization strategy (Hammond-Kaarremaa, 1994; Thomas & Harri-Augstein, 1977).

The most common organization of material, especially in the face-to-face classroom, is to provide material in a linear fashion progressing from the most basic material to the most complex. This organization strategy, a serialist one, works very well for sequential students. As global students prefer a framework-style organization, context is useful to help these students learn. Thus, global students may be reached by providing an overview or goal at the start of lecture or material to provide the needed context. Global students prefer to find their own way to solve problems; therefore, this should be encouraged.

2.2.1.5 Validity and Reliability of the Index of Learning Styles

There are several types of validity (e.g., criterion, content, concurrent, construct, predictive, etc.); however, the analysis of ILS in literature (Cook & Smith, 2006; Felder & Spurlin, 2005; Genovese, 2004; Litzinger, Lee, & Wise, 2005) focuses on *construct validity* (Cronbach & Meehl, 1955) which is a qualitative description on how well the instrument truly measures the intended phenomenon. For example, if a written instrument intended to measure a particular trait uses complex words or phrases, it might be measuring reading comprehension instead of the desired trait and would have low construct validity. Construct validity is evaluated through the analysis of evidence.

The ILS is argued to be valid if it identifies consistencies and differences in learning styles based on what is known about learning styles (Cook & Smith, 2006; Litzinger et al., 2005). Both studies point out that students are expected to have some consistency in learning styles within a faculty and differences in learning styles between different faculties. Litzinger et al (2005) add that there are no expected gender differences in the learning styles identified for students. Several studies (Cook & Smith, 2006; Felder & Spurlin, 2005; Genovese, 2004; Litzinger et al., 2005; Lopes, 2002) confirm that the ILS identifies learning styles with consistency within faculties and identifies differences between faculties as expected. For example, it is seen that engineering students across several universities tend to have consistent learning styles (Felder & Spurlin, 2005). Litzinger et al. (2005) examined students from the education, engineering and liberal arts faculty in two colleges for any differences. They found that there were significant differences between engineering, liberal arts and education students while each faculty showed some consistency. Lopes (2002) similarly examined students from the sciences and humanities and found some differences in learning styles between faculties, while being similar internal to the faculty. Litzinger et al (2005) identified the learning styles of male and female engineering students and found no significant difference in learning styles as expected. All of this evidence suggests that the ILS is performing as expected and so is a valid instrument for identifying learning styles.

There are also several different types of reliability, with two being evaluated in literature for the ILS: *test-retest reliability* (Cook & Smith, 2006; Felder & Spurlin, 2005; Livesay, Dee, Nauman, & Hites Jr, 2002; Seery, Gaughran, & Waldmann, 2003) and *internal consistency reliability* (Bacon, 2004; Cook & Smith, 2006; Genovese, 2004; Zywno, 2003).

Test-retest reliability measures whether an assessment instrument will repeatedly make the same measurement on a series of tests. For example, a scale which measures a 1 kg weight consistently as 1 kg would be said to be very reliable; whereas, if it measures the weight as 1 kg, 0.5 kg and 1.5 kg on three tests it would be very unreliable. For psychometric instruments, test-retest is typically done by testing a participant and then retesting them 1 or more times after a significant period of time. The period of time needs to be long enough so that the participant will not answer similarly from memory, but not so long that the trait to be evaluated may have changed from other factors. To be considered reliable, the measurements should have significant correlation. Internal consistency reliability applies only to assessment instruments with multiple test items (such as the ILS) and measures the degree to which the different test items measure the same trait.

Table 2. Test-retest correlation coefficients for Index of Learning Styles

Study	Time Frame	Test-Retest Correlation Coefficient				N
		A/R	S/I	V/V	S/G	
Seery et al. (2003)	4 weeks	0.804*	0.787*	0.870*	0.725*	46
Cook & Smith (2006)	3 months	0.809**	0.856**	0.703**	0.651**	89
Livesay et al. (2002)	7 months	0.73***	0.78***	0.68***	0.60***	24
Zywno (2003)	8 months	0.683*	0.678*	0.511*	0.505*	123

* p<0.01, ** p<0.0001, *** p<0.05

The ILS has been evaluated for test-retest reliability with engineering students at three different time frames, 4 weeks (Seery et al., 2003) , 7 months (Livesay et al., 2002) and 8 months (Zywno, 2003), and with medical students with a 3 month gap between tests (Cook & Smith, 2006). The results from these four trials are shown in Table 2. Felder and Spurlin (2005) state that the four week interval is the ideal time period and that the high correlations in addition to the statistical significance of the two other results suggest that the ILS is “satisfactory” with respect to test-retest consistency. Felder and Spurlin did not consider the study by Cook and Smith as it did not exist at the time; however, the results from Cook and Smith only strengthen their conclusion that the ILS is satisfactory with respect to test-retest reliability. Furthermore, Cook & Smith show that test-retest reliability for the ILS may extend beyond the engineering faculty.

Internal consistency is evaluated using *Cronbach’s alpha* (α) (Cronbach, 1951) which examines the relatedness of each possible pair of test items. Cronbach’s alpha is based on a method of assessing consistency used prior to Cronbach’s alpha called split-half. When using the split-half method, the instrument is split in half and evaluated as if it were two separate instruments, thus providing two measurements. If the measurements from each half of the instrument are similar then the instrument is considered consistent. The drawback to the split-half method is that it is dependent on what items are selected for each half and selecting different items will result in a different measurement of consistency (Brownell, 1933). Cronbach, in defining alpha, proposed to correct this problem by considering all the possible split-halves. Cronbach’s alpha, shown in Formula 1 (Cronbach, 1951), is a function of the number of test items (n), the variance of the measurement scores (V_t) and the variance for each item (Vi).

$$\alpha = \frac{n}{n-1}\left(1 - \frac{\sum_{i=0}^{n} V_i}{V_t}\right) \tag{1}$$

For instruments which assess an attitude (or preference) an $\alpha > 0.5$ is considered acceptable (Tuckman & Harper, 2012). Several researchers have calculated Cronbach’s alpha for the ILS in the context of different faculties with the results shown in Table 3.

Table 3. Cronbach's Alpha values by learning style dimension for different faculty

Study	Context	N	A/R	S/I	V/V	S/G
Bacon (2004)	Business	161	0.60	0.70	0.66	0.47
Cook & Smith (2006)	Medical	89	0.62	0.77	0.72	0.65
Genovese (2004)	Education, Psychology	131	0.63	0.72	0.71	0.53
Litzinger et al. (2005)	Engineering, Liberal Arts, Education	572	0.60	0.77	0.74	0.56
Livesay et al. (2002)	Engineering	242	0.56	0.72	0.60	0.54
Spurlin (Spurlin, 2002)	Unspecified	584	0.62	0.76	0.69	0.55
Van Zwanenberg et al. (2000)	Engineering, Business	284	0.51	0.65	0.56	0.41
Zywno (2003)	Engineering	557	0.60	0.70	0.63	0.53

Based on their individual results, the literature agrees that the A/R, S/I and V/V dimensions are internally consistent. One issue raised repeatedly in literature is a high degree of correlation (low orthogonality) between the S/I and S/G dimensions (Cook & Smith, 2006; Genovese, 2004; Zywno, 2003), suggesting that they are not unique traits. In particular the S/G dimension is questioned as the results are very close to the acceptable limit, and in the case of the studies by Bacon (2004) and Van Zawnenberg et al. (2000) slightly below. Felder & Spurlin (2005) respond to this criticism by pointing out that this is only an issue from a psychometric perspective. The purpose of the FSLSM (and by extension the ILS) is to measure these traits with the aim towards providing guidance to teachers and students. Thus, to account for different preferences in the S/I and S/G dimensions, different approaches are required with respect to course design, teacher interventions and decisions made by students about their own learning.

2.3 Working Memory Capacity

This section discusses WMC beginning with an overview of its origins. This is followed by a look at cognitive load theory (CLT) and its implications on learning and curriculum development. Afterwards, the techniques for identifying WMC are examined, with some focus on the OSPAN test as a computerized form of this test, WebOSPAN (Lin, 2007), is used by this research to identify students' actual WMC. Lastly, this section concludes with an examination of the validity and reliability of the OSPAN test and computerized versions of OSPAN.

The origins of WMC come from the investigations of Miller (1956) into the *span of absolute judgement* (the amount of information that can be received simultaneously) and the *span of immediate memory* (the number of items that may be retained at a time, i.e. WMC). The limit for the span of absolute judgment was found by examining the data from other studies on the number of unique items that could be identified by a person at a time, such as pitches (Pollack, 1953), tonal intensities (loudness) (Garner, 1953) and tastes (Beebe-Center, Rogers, & O'connell, 1955). Miller found that by determining the number of bits required to describe the maximums for each of these different tests, that the span of absolute judgment was limited by the total amount of information received. This limit of span of immediate memory was investigated by examining how many binary digits, decimal digits, letters, letter and digits and mono-syllable words could be retained in memory at a time by a person (Hayes, 1952). The limit for immediate memory was found to be more dependent on the number of items, than the size (in bits) of each item and the limit was found to be between 7±2 items (Miller, 1956).

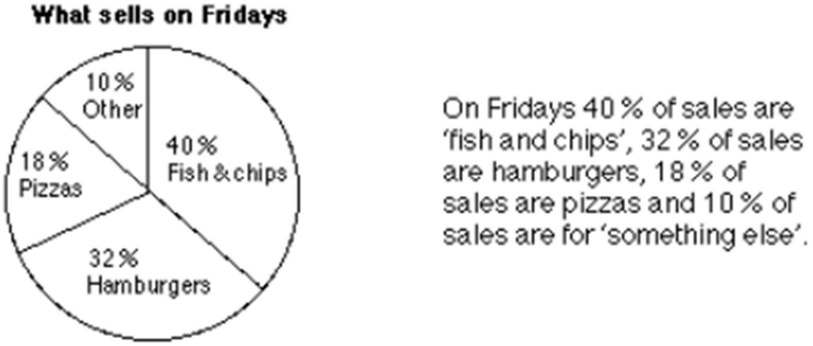
Knowing that WMC is limited, Sweller (1988) developed cognitive load theory (CLT) to explore the effects of cognitive load on learning. At the time, Sweller (1988) states that the only means of measuring problem difficulty was to present the problem to the student and measure the outcome. This means that if a problem is found to be inappropriate for a specific student, this discovery is made too late for that student.

Sweller's motivation was to find a mechanism for measuring problem difficulty, and in turn its effectiveness as a learning tool, for students without needing to present the problem to the student. Similar to Felder and Silverman argument that traditional curriculum may not be effective for all students, Sweller (1988) argues that curriculum which has a high cognitive load, for example problem solving in mathematics, may not be an effective learning tool for all students.

An extension to WMC was presented in the work by Mayer and Moreno (1998) in which they present evidence that WMC exists as two separate channels, visual and auditory. Furthermore, they argue that cognitive overloading can occur on either channel independently. For complex tasks, such as learning, when WMC is overloaded additional mental effort is required (Kane & Engle, 2000) and causes an increased number of errors, time or a reduction in transfer of learning (Cooper, 1998; Kirschner, 2002; Van Merriënboer et al., 2002). The literature review found that most research has been focused on avoiding cognitive overloading; however, underloading also reduces student performance as the mind is under stimulated (Paas et al., 2004; Teigen, 1994). Thus, to obtain peak performance requires putting the student under the optimal cognitive load, neither too low nor too high. The early literature on CLT looks at considering curriculum broadly; however, Mayer and Moreno (2003) argue that in part curriculum needs to be individualized to each student. What follows is a closer examination at some of the effects poorly designed curriculum can have on WMC and how to avoid such effects.

One study at the University of New South Wales examined using several techniques for avoiding effects which may increase cognitive load: goal free effect, worked example and problem completion effect, split attention effect, redundancy effect and modality effect (Cooper, 1998). Cooper recommends the following guidelines when designing learning material. First, problems should be goal free which is a design paradigm where students are not told to solve a specific problem but rather to “find what they can” (Cooper, 1998). The goal free problem design forces students to examine the problem data and work incrementally forward and has been found to reduce cognitive load (Ayres, 1993). Second, Cooper encourages more use of worked examples where students are shown how to solve a problem in a step-wise fashion. This reduces cognitive load as only small amounts of information are required to be processed at a time, the problem state and the transition rule being taught for the step (Paas, 1992). Third, LOs which require both visual and verbal elements should be integrated together so as to not have student split their attention between them (Chandler & Sweller, 1992). Fourth, when both visual and verbal content is integrated caution should be taken not to have it repeated simultaneously as this needlessly increased the amount of information to be processed. Figure 3 shows an example of redundant information. Additionally the pictured graph on the left is a better choice of content, if it were alone, as it integrates both the visual and verbal elements. As working memory exists as two separate dedicated channels, visual and verbal (Baddeley, 1992; Mayer & Moreno, 1998) and the load on each channel is independent, the fifth guideline is to have both visual and verbal elements in LOs when possible. This will allow for an effective increase in usable WMC as the student will use both channels.

Figure 3. Example of redundant information and visual / verbal integration (Cooper, 1998)



On Fridays 40 % of sales are 'fish and chips', 32 % of sales are hamburgers, 18 % of sales are pizzas and 10 % of sales are for 'something else'.

Another study (Mayer & Moreno, 2003) makes similar recommendations as Cooper (1998); however, they have three additional recommendations. First, they suggest a segmentation effect where material is both divided into small segments and the student is allowed to control the movement between segments. Their experiment consisted of dividing content into 16 segments and presenting it to sets of students. The control group was shown the material in a continuous sequence while for the experimental group the sequence was paused after each segment until the student hit a "continue" button. They found that the experimental group had a higher amount of learning transfer over the control group. They suggest that prior to the main educational content, pre-training on important terms will aid in reducing cognitive load (pre-training effect). Furthermore, they suggest that extraneous material be removed from LOs as this simply increases the cognitive load to little benefit (coherence effect).

As a first step towards optimizing cognitive load, WMC for a student must first be identified. Classically, identifying WMC has been done with dedicated tests using concurrent processing tasks where the participant must perform a memorization task in addition to some other task that varies from test to test. One early such test is the reading span task (Daneman & Carpenter, 1980) where the participant reads a series of sentences and must memorize the end words of each sentence. After reading the series of sentences, the student must recall and state the end words. Turner and Engle (1989) developed the operation span task (OSPAN) task while investigating the possibility that the reading span task was measuring reading comprehension rather than WMC by evaluating the independence of several different span tasks: sentence-word span (reading span), sentence-digit, operation-word (OSPAN), operation-digit, simple word and simple digit. Sentence-word and sentence digit require the participant to read a sentence and memorize the end word or a digit placed at the end of the sentence respectively. Operation-word and operation digit replace reading a sentence with solving the truth value of a mathematical expression (e.g. “$8 \div (2 + 2) = 4$?”). Simple word and simple digit require the participant to simply memorize a series of words or digits, i.e. there is no concurrent processing for these tasks while the first four all contain concurrent processing. Of the six tasks, the measured WMC for the sentence-word span task, sentence-digit and operation-word were found to correlate well ($r > 0.5$, $p < 0.0001$) showing that WMC measurement was task independent and that the sentence-word span does not simply measure reading comprehension although higher WMC does predict better reading comprehension (Daneman & Merikle, 1996). Literature shows that both the reading span (DeCaro, Peelle, Grossman, & Wingfield, 2016; McVay & Kane, 2012) and OSPAN tasks (Chang et al., 2013; Lin, 2007) remain in common use in research; however, the remainder of this section will focus on the validity and reliability of OSPAN as a computerized version is used by this research to measure students’ actual WMC.

2.3.1 Validity and Reliability of OSPAN

Much research has been done on the validity and reliability of OSPAN. Literature shows that the classical OSPAN is valid and reliable (Conway, Cowan, Bunting, Therriault, & Minkoff, 2002; Engle, Tuholski, Laughlin, & Conway, 1999; Klein & Fiss, 1999). Furthermore, the automatic computerized version used by this research, WebOSPAN, has also been found to be reliable (Lin, 2007).

The validity of OSPAN is based on latent variable analysis comparing the measurements of several cognitive traits, such as general intelligence, short term memory and processing speed, to the measurement of WMC (Conway et al., 2002; Engle et al., 1999). Since participants who are measured as having high WMC by OSPAN are also identified as having higher fluid intelligence, short term memory and processing speed (as expected) this suggests good construct validity for OSPAN.

With respect to reliability, Klein and Fiss (1999) used a test-retest methodology to examine the reliability and stability of OSPAN by conducting an initial test, a retest after 3 weeks and a second retest after an additional 6 or 7 weeks. The lowest test-retest correlation was 0.66 between the initial test and the 2nd retest. Using Heise's (Heise, 1969) formula for reliability (which takes into account measurement errors) they found that OSPAN had a true reliability of 0.883. From this, they conclude that OSPAN is "an extremely reliable measure" (Klein & Fiss, 1999). They examined the internal consistency using Cronbach's Alpha (Cronbach, 1951) and got results of 0.776, 0.810 and 0.829 for the initial test, retest and 2nd retest respectively, which is considered acceptable to good (Kline, 2013; Tuckman & Harper, 2012).

Lin (2007) examined the validity of WebOSPAN by calculating the Pearson Correlation Coefficient and 2-tailed significance between two metrics (OpTotal and

SetSize) measured by WebOSPAN (other metrics are gathered by Lin and examined but they do not relate to the discussion on validity). OpTotal is the total number of words recalled correctly across all series where all of the words in a series must be recalled correctly and in the proper order (i.e. this is the measurement of WMC as defined by the operation span task (Turner & Engle, 1989)). SetSize is the largest word series size correctly identified by the participant. The argument is that for a reliable instrument participants should be consistently right or wrong based on the maximum series size they can successfully perform, i.e. these two metrics should have high correlation. This same argument is used by both Engle et al. (1999) for OSPAN and by De Neys (2002) for GO-SPAN (another computerized OSPAN task). The correlation coefficient is found to be 0.811 with a 2-tailed significance of 0.01, thus showing that WebOSPAN is reliable.

2.4 Survey of Other Automatic Approaches for Identifying Learning Styles

This section examines several other automatic approaches for identifying learning styles. First, approaches which use a literature-based method are examined, followed by approaches which use a data-driven method with AI/CI algorithm(s). A summary of the approaches examined are shown in Table 4.

Latham et al. (2012) propose a literature-based approach used with a natural language conversational agent called Oscar. In their approach, logical rules are derived from literature, for example if a student answers a question correctly after being shown an image they may have a visual preference. The students' behaviors are extracted from the dialogs between the agent and student and the rule set is applied to the data to identify the learning styles. The experimental data showed a precision value of 86% for A/R dimension, 75% for S/I, 83% for V/V and 72% for S/G; however, a significant drawback of this approach is that it is tied to the Oscar system and cannot be generalized.

Table 4. Summary of automatic approaches to identify learning styles

Study	Algorithm	Number of participants in the evaluation	Limitations
Latham et al. (2012)	Rules	75-95	Non-generic
Graf et al. (2009)	Rules	75	-
Garcia et al. (2007)	Bayesian network	77	Cannot identify V/V dimension
Carmona et al. (2008)	Bayesian network	No evaluation	Students must rate LOs
Özpolat and Akar (2009)	Naïve Bayes decision tree	40	-
Cha et al. (2006)	Decision tree Hidden Markov model	23-49	Identifies subset of students only
Dorça et al. (2013)	Reinforcement learning	Simulated data	-
Villaverde et al. (2006)	Artificial neural network	Simulated data	Cannot identify V/V dimension

Another literature-based approach called DeLeS (Graf, Kinshuk et al., 2009), derived relationships between behavior patterns and learning styles. They used this information to construct rules to produce hint values. The learning style is calculated by using an unweighted average of the hint values. They report a precision of 79% for A/R, 77% for S/I, 77% for V/V and 73% for S/G. Unlike Oscar, the behavior patterns used by DeLeS are general to any LMS.

Garcia et al. (2007) used a Bayesian network (BN) in order to detect students' learning styles. First, they identified the behaviors that may be relevant to identifying learning styles, such as whether the student participates in forums or revises exams. The initial probabilities for the BN are based on expert knowledge. They then used the ILS

(Felder & Solomon, 1998) to identify the learning styles for 50 real students. These students then use the learning environment and their behaviors are used to further train the BN to identify their learning styles (based on knowing their actual learning styles). The trained BN was then evaluated using 27 real students. With these students the belief state of the BN is updated as they interact with the learning system, with the BN providing a probability that the student has a particular preference. This probability is mapped to the FSLSM scale to identify the strength of the student's preference. The example they provide is if a student has a 75% probability of a preference for a learning style they consider this a value of 7 on the FSLSM scale (which is 1,3,5,7,9,11) for that dimension. In evaluating their algorithm, they introduce a similarity metric (SIM) in which the FSLSM is divided into three regions, neutral (or balanced) and two poles. The SIM returns 0, 0.5 or 1 as a function of the region of the student's actual and identified learning style (1 if they are the same region, 0.5 if they are adjacent, 0 if they are opposed). Using the SIM metric, they obtained a precision of 58% for A/R, 77% for S/I and 63% for S/G (the V/V dimension was not considered).

Carmona et al. (2008) also used a dynamic Bayesian network to identify learning styles. To LOs in the system they associated five learning style relevant attributes: format (image, text, etc.), resource type (exercise, example, etc.), interactivity level (very low to very high), interactivity type (active, expositive or mixed) and semantic density (very low to very high). Each of the attributes is mapped to one or more FSLSM dimension(s). Every time a student selected a LOs, they would be asked to rate the usefulness of the material from 1 to 4. After rating the LO, the rating is used as evidence to adjust the belief state of the network. The drawback to this approach, when compared to other automatic approaches, is that it requires input from the student instead of working solely on their behaviors. This is potentially intrusive for the student and there is no guarantee that the student will rate solely based on how it appealed to their learning styles. They did not evaluate their approach.

Özpolat and Akar (2009) examined the LOs selected by students as being most useful in response to a keyword search. Using a naïve Bayes decision tree (NBTree) the keyword attributes of the LOs are converted into learning styles identification for the student. For example, if the student selects LOs with the keyword attributes such as graphs, charts or jpg then the student is more likely to be classified as having a visual preference. They report a precision of 70.0% for A/R, 73.3% for S/I, 73.3% for S/G and 53.3% for V/V.

Cha et al. (Cha et al., 2006) evaluate two approaches in their study, decision trees and hidden Markov models (HMM). For building a decision tree they gathered numerous behaviors (they state 58 but they are not fully listed) such as the number of clicks on particular icons, time spent on some activities, quiz grades and reading or posting to forums. They then identify the learning styles for 70 real students using the ILS (Felder & Solomon, 1998) and then eliminate any student with a preference between 1-3 on the FSLSM scale, i.e. a balanced preference. A subset of the data is used to train the decision tree based on knowing the actual learning styles. For the HMM-based approach is trained to recognize the sequence of buttons clicks from students with known learning styles. The HMM is used to identify a future student's learning styles from their click sequence. For the decision tree, they found a precision of 66.7% for A/R, 77.8% for S/I, 100% for V/V and 71.4% for S/Q. The HMM approach is reported as having precision values of 66.7%

for A/R, 77.8% for S/I, 85.7% for V/V and 85.7% for S/Q. Although some of the results are quite high, as mentioned above they excluded all students with a balanced preference. Thus, their approach has a major limitation in that it can only identify students with a strong preference one way or another, i.e. it cannot identify students with a balanced preference.

Dorça et al. (2013) presented an approach for identifying learning styles using reinforcement learning (Q-learning). With this approach, they assume an a priori probability of a student having a learning style in accordance with the meta-study by Felder & Spurlin (Felder & Spurlin, 2005). They associate LOs in the system to particular learning styles based on expert knowledge. The student is then presented LOs to achieve a learning goal followed by an assessment on how well they learned the material. The probability that the student has particular learning styles is then reinforced based on the performance assessment and how well the learning styles of the LO match the students' current predicted learning styles. They considered three different strategies towards reinforcement, either reinforcing for high performance only, low performance only (inverse reinforcement) or both. So for example, with a reinforcement with high performance strategy if the student performs well on an assessment it is more likely they have the learning styles associated with the LO. They evaluated their approach using simulated data.

Villaverde et al. (2006) evaluated training a feed forward ANN (a 3-layer perceptron) with backpropagation under a supervised learning model. They used 10 behavior patterns as inputs such as what kind of reading material did the student prefer, does the student revise exams prior to submission and does the student ignore, post or read forums? They translate the behavior patterns from a qualitative value to real values from -5 to +5 (e.g., the students make few exam revisions is translated to -5). Their approach provides an integer output from +1 to -1. They translate value of -1 to active, intuitive or sequential and +1 as reflective, sensing or global (the V/V dimension is not considered). They report an average precision of 69.3% across the three dimensions they considered using simulated data (they do not report individual precision values for each dimension). The approaches in this research which use ANNs (LSID-ANN and LSID-SISO) are similar to this approach in that they use a 3-layer perceptron with behavior patterns as inputs; however, there is a notable difference. First, rather than using a single ANN to identify three learning style dimensions simultaneously, LSID uses a single algorithm for each learning style dimension. By doing so, LSID-ANN and LSID-SISO are able to find a globally optimal solution for each learning style dimension as opposed to the solution that has the best average across the learning styles. Also, building a separate approach for each learning style dimensions allows the behavior inputs to be only those expected to be relevant for that learning style dimension, as opposed to using all of the behavior patterns for all of the learning style dimensions.

2.5 Survey of Other Automatic Approaches for Identifying Working Memory Capacity

The only automatic approach found to identify WMC is "Detecting Working Memory Capacity" (DeWMC) (Chang et al., 2013). DeWMC works by creating rules to generate hint values, which are then averaged to give the student's WMC. The rules are based on the relationships between navigational patterns when using an LMS and WMC and the relationship between learning styles and WMC. DeWMC was evaluated using behavior data, learning styles as identified by the ILS (Felder & Solomon, 1998) and compared it to WMC as identified by WebOSPAN (Lin, 2007) and found to have a precision of 80.9%.

2.6 Background on CI Algorithms

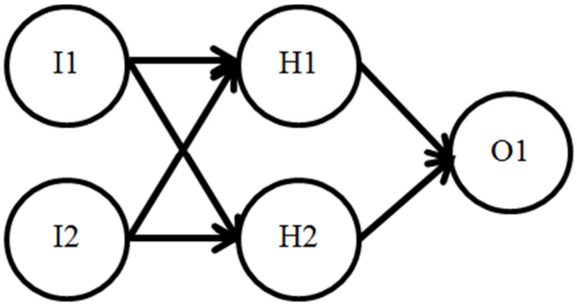
This section will examine each of the CI algorithms used in this research. The underlying basis of each algorithm and how they are used to solve problems will be discussed. Additionally, any algorithm specific control parameters will be discussed.

2.6.1 Artificial Neural Networks

ANNs has been described as a "universal approximator" (Hornik, Stinchcombe, & White, 1989) as it is an algorithm which selects from a set of hypotheses (or functions) one which best fits the data samples (Mitchell, 1997b). In so doing, it allows future data samples to be properly identified. ANNs are inspired by the cellular neurobiology of the brain, the most advanced mechanism for intelligence to human knowledge. The building block of the brain is the neuron, a cell which at a basic level consists of dendrites, soma (cell body) and axon and functions as follows (Zigmond & Bloom, 1999). Dendrites are incoming connections carrying electrical impulses from other neurons. If the cumulative effect of the incoming impulses raises the electrical charge on the soma above the threshold of excitation, the neuron becomes activated and sends out an electrical impulse to its axon which connects to the dendrites of other neurons.

An ANN is composed of a graph of virtual neurons (henceforth, just neuron). Each neuron is classified as an input neuron, output neuron or hidden neuron. Figure 4 shows a simple ANN consisting of two input neurons (I2 and I2), two hidden neurons (H1 and H2) and a single output neuron (O1). The neuron, much likes its biological counterpart, is composed of three basic elements: input links, output links and an activation mechanism. Each of these is described as follows from the work of Mitchell (1997b).

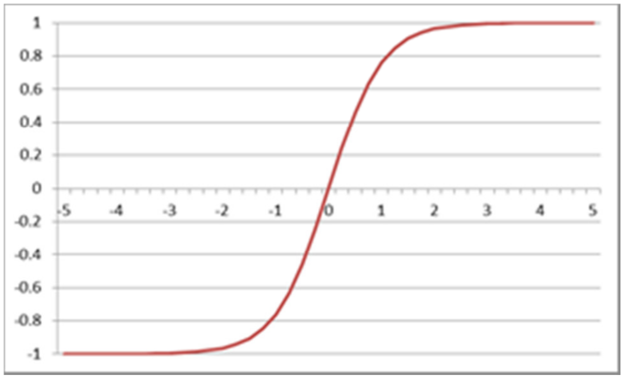
Figure 4. A Simple ANN (a 3-layer perceptron)



The links, as a connection between two neurons, act as both inputs and outputs. For example, in Figure 4, the link (L_I1_H1) between I1 and H1 acts as an input to H1 and an output for I1. Each link has a weight and strength associated to it with the weights having real values from 0 to 1 and strength values as real values typically bounded to ±1. The strength value of a link is the output value of the neuron. So, the strength of the link L_I1_H1 is the output value of the neuron I1. The output value of a neuron is determined by the inputs and the activation mechanism as described next.

The activation mechanism consists of a transformation function and an optional activation threshold. The transformation function converts the total strength applied to the neuron into an activation value where the total strength applied to a neuron is the weighted sum of the strength of the neuron's input links. The exception is the strength of an input neuron, which has no inputs links, is the value of a data input. If the activation value is above the optional activation threshold then the neuron is said to have fired. When a neuron fires, the activation value is used as the output value which is used as the strength value for its output links. Although any function may be used for the transformation function, generally a sigmoid function is used, such as tanh. A sigmoid function has a distinct s-shaped curve and is typically bounded to ±1 (shown in Figure 5).

Figure 5. S-shaped curve of the function tanh



There are many different topologies for ANNs. One common topology, and the one used in this research, is the multilayer perceptron (MLP). The MLP consists of two or more layers of neurons; however, three layers is typical, an input layer, hidden layer and output layer (Mitchell, 1997a) as shown in Figure 4 above. The topology of the MLP is such that each input neuron is connected to each hidden neuron and each hidden neuron is connected to each output neuron. Each input neuron has one input link and similarly each output neuron has a single output link. In this configuration, as there are no cyclical connections, the information feeds forward from the inputs to the outputs.

To train an ANN, both a learning model and training method must be selected. The three learning models used by CI algorithms are supervised, unsupervised and reinforcement learning (Mitchell, 1997b). Used by this research is the supervised learning model, where an error is calculated from comparing the ANN's output to an expected value. Training is continued iteratively until the error is minimized. The supervised learning model is only usable when the training data contains the expected value, which is true for this research as the student data contains the students' actual learning styles and WMC. The recommended method for training a MLP under the supervised learning model is backpropagation which functions as follows (Mitchell, 1997b).

Backpropagation works by processing the MLP in reverse and adjusting the weights in the neural links so as to reduce the error. The formula for finding the weight adjustment is called the *delta rule* and is derived from the gradient descent algorithm (Mitchell, 1997b). The gradient descent algorithm is a process of iteratively altering the value of x for a function, f(x), to find the minimum value of f(x). Each iteration the value x is modified in the descending direction of f(x) and by a step size in proportion to the negative gradient of the function. Since, only a step along this gradient is required, a parameter is needed to control the step size and is called the learning rate ($0 \leq \eta \leq 1$). The learning rate is generally kept low to prevent oscillation over the minimum; however, if it is too low the training process can become prone to being stuck in local minima.

One issue with the backpropagation algorithm is that it steps towards the local minimum which may not be the global minimum. Assuming no overfitting is occurring, the ideal is for the MLP to be trained to the global minimum error. The momentum ($0 \leq m \leq 1$) control parameter is used to push the MLP out of local minima; therefore, exploring the weight solution space more fully (Mitchell, 1997a). Momentum adjusts the weight adjustment formula by adding a portion of the previous weight adjustment to the current weight adjustment. Like learning rate, momentum is generally kept quite low as a high momentum makes it very difficult for the training process to come to rest in the vicinity of the global minimum once it is found.

There are two training modes available for MLPs: individual and ensemble (Mitchell, 1997a). Although backpropagation is done following each sample, it is not necessary to immediately apply the weight modifications to the links. If the weight modification is applied immediately, this is called individual training mode. Otherwise, after each generation the weight adjustments from each sample may be summed and then applied, and this is called ensemble training mode. The advantage of the individual training mode is that more steps are taken more quickly, while the advantage to the ensemble mode is that each sample is processed by an identical MLP. Which training mode is best is problem specific.

2.6.2 Ant Colony System

ACS is one of the more recent versions in the family of ant colony optimization (ACO) algorithms (Dorigo & Gambardella, 1997b; Dorigo & Stützle, 2010). The inspiration for all of the ACO algorithms is real ant behavior when foraging for food. Ants share information by laying pheromone along the path to food sources. Suppose there are two paths to a food source, one long and one short, since the ants have no information about the length of the paths an equal number of ants will choose the long and short paths. The shorter path will have a higher density of ants and so pheromone will accumulate quicker on the short path. This will encourage more ants to the shorter path and so the most food will be gathered for the least effort. Thus, for the ants, the shorter path is the higher quality solution.

ACS uses a population of artificial ants (henceforth just ants) to search a solution space for optimal solutions by using a pheromone-like mechanic (Dorigo & Gambardella, 1997b). ACS was originally proposed to solve the problem where a travelling salesman must travel to a set of cities while travelling the shortest distance. ACS has since been used to find optimal solutions for other problems such as quadratic assignment problem (Gambardella, Taillard, & Dorigo, 1999), job scheduling (Rajendran & Ziegler, 2004) and economic dispatch (Pothiya, Ngamroo, & Kongprawechnon, 2010).

To use ACS, the solution space must be described as a graph. The ants are then inserted into a node and set to traverse the graph by iteratively choosing a link to follow. The graph must be crafted such that a completed path may be decoded into an appropriate solution. For example, for the travelling salesman problem the nodes may represent the cities, the links represent the roads connecting the cities and the completed path is the route taken by the salesman. Whereas, for the scheduling problem, each node could represent a job and the links a choice of which job to schedule next and the path may then be decoded as a job schedule.

Each link in the graph has two associated values: local quality (l) and pheromone (global quality) (τ). The overall quality (Q) of the link is a weighted sum of these two values, with weights α and β respectively (shown in Formula 2). To choose a link, the ants use a pseudo-random proportional rule which encourages exploitation of higher quality links. At each node, a random value from 0 to 1 is selected and if this value is less than the control parameter, exploitation factor (q_0) then the highest quality link will be followed; otherwise, a random link will be selected with a preference for higher quality links. When selecting a random link, not all links will necessarily be considered at each node. Each ant keeps a *tabu list* of nodes which may not be visited. Typically this is done to prevent revisiting of nodes but may be for any problem specific purpose. Using expert knowledge, an optional *candidate list* of preferred links may be pre-generated for each node. An ant will consider only links on the candidate list unless all of them are on the tabu list, in which case it will consider all possible links not on the tabu list. Selecting a link is then done using a roulette wheel selection where the odds of selecting a link (*S*) are equal to the quality of the link divided by the sum of the quality of all permissible links from that node as shown in Formula 3. This process continues until the ant can no longer move or has built a complete solution at which point its path is decoded into the candidate solution.

$$Q_i = \alpha \times l_i + \beta \times \tau_i \tag{2}$$

$$S_i = \frac{Q_i}{\sum_{j=1}^{N} Q_j} \tag{3}$$

Two rules, the *global pheromone updating rule* and *local pheromone updating rule* control the pheromone values on each link. The global pheromone updating rule, although a common feature of ACO algorithms, may have some variations between them (Dorigo & Stützle, 2010). As only ACS is used in this research only the global pheromone updating rule for ACS is described next, followed by the local pheromone updating rule for ACS.

The global pheromone updating rule occurs after each iteration and consists of two mechanisms. The first mechanism accumulates pheromone on the links as follows. Each solution generated by the ants is evaluated using the problem's fitness function (*F*). If the fitness of a solution is better than the best solution found so far, then the solution is recorded as the global best solution (s_{gb}). Either the links along iteration best solution (s_{iter}) are updated or more commonly the s_{gb} is reinforced (Dorigo & Stützle, 2010) by 1/F(s). The second mechanism of the global pheromone updating rule reduces pheromone on the links, as if pheromone values were to only increase then it would become increasingly harder for the ants to explore for new solutions. Thus the colony needs to forget some information which is done by reducing pheromone values. After each iteration (t), every link in the graph has its pheromone value reduced as a function of the current pheromone times the evaporation rate (ρ) (shown in Formula 4) (Dorigo & Gambardella, 1997b).

$$\tau_i^{t+1} = (1 - \rho)\tau_i^t \tag{4}$$

The local pheromone updating rule occurs when an ant traverses a link (Dorigo & Gambardella, 1997b). The rule uses a consumption mechanism to reduce pheromone on links. In this way, ants which follow will be less likely to select the same path and so be more likely to explore the solution space. Pheromone on the link is reduced as a function of the pheromone on the link and the consumption rate (τ_0) (shown in Formula 5).

$$\tau_i = (1 - \tau_0)\tau_i' \tag{5}$$

2.6.3 Genetic Algorithm

Genetic algorithm (Mitchell, 1998) is a combinatorial optimization algorithm inspired by concepts from evolution. In biological terms, as successive generations of parents mix genes when producing offspring over time the offspring become fit to their environment. Thus, GA operates under the premise that by re-combining the building blocks of solutions that over successive generations better solutions will be found, ultimately leading towards a globally optimal solution.

A population of size P genomes is produced and each genome is composed of N genes. Gene values are very flexible and may be expressed as a bit, integer or real value, although not necessary typically non-bit representations are bounded. Each gene represents an element to the solution such that the entire genome represents, once decoded in a problem specific way, a solution. Each genome is assigned a fitness value using a problem specific fitness function.

After assessing the initial population, the main processing loop of the algorithm is started which consists of the following steps for each generation: selection, crossover, mutation and survival. During the selection step pairs of genomes are selected from the population. There are numerous ways this can be done; however, most techniques have the similar guiding principle of preferring more fit members. One technique, and the one used in this research, is roulette wheel selection. Roulette wheel works by assigning each genome (G_i) the chance of being selected (S_i) equal to its fitness value (F_i) divided by the sum of all fitness values in the population as shown in Formula 6. A random value is selected from 0 to 1 and the appropriate genome is selected. For example, if there are four genomes (G_1, G_2, G_3, G_4) with the odds of selection (0.5, 0.3, 0.15, 0.05) respectively if the random value selected is 0.6 then G_2 is selected as it is greater than the odds for G_1 but less than the odds of $G_1 + G_2$. During the selection process the same genome may be selected more than once; however, the same pair of genomes is not permitted to be selected in the same generation.

$$S_i = \frac{F_i}{\sum_{j=1}^{P} F_j} \tag{6}$$

After all of the pairs have been selected, the crossover operator is applied to each pair. As with the selection process, there are several techniques which may be used with some more common approaches being uniform crossover, one point crossover and two point crossover. For this research, uniform crossover is used and functions as follows. Uniform crossover assigns each gene an identical (hence uniform) chance, called the crossover weight ($0 < C < 1$), of being swapped. For each gene a value from 0 to 1 is randomly picked, and if this value is less than C the genes are swapped. Since it is based on random chance, there may be zero to N swaps, with zero and N swaps being non-ideal as the result is the same genomes (this issue is resolved in the mutation step described next). Crossover weight is generally kept relatively high to promote a relatively large number of new genomes, although this can be disruptive of good gene combinations. The crossover operation fulfills the premise of the GA by seeking to exploit previously found good solutions and using their genes to build new solutions. An example of the crossover operation is shown in Figures 6 and 7. This example assume each genome (A & B) has 4 genes called A1 to A4 and B1 to B4 and that the crossover operation will be done between A2 and B2.

Figure 6. Example of the crossover operation, A2 swapped with B2 (pre-swap)

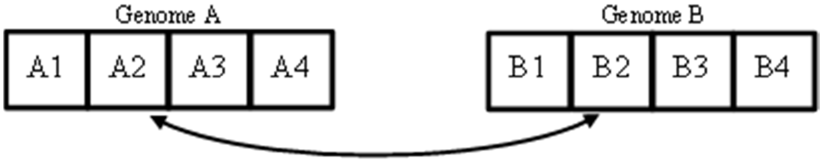
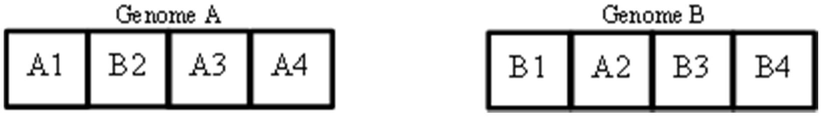


Figure 7. Example of the crossover operation, A2 swapped with B2 (post-swap)



The mutation step is intended to ensure that particular gene values do not become utterly dominant in the population, thereby causing stagnation and a lack of exploration. The mutation step operates by possibly selecting one or more genes in each genome produced by the crossover step and changing it to a random value. Much like the uniform crossover operation, each gene is given the identical chance to be mutated called mutation weight ($0 < M < 1$). For each gene, a random value from 0 to 1 is picked, and if the value is less than M the gene is mutated. As with the crossover step, there may be zero to N mutations. If there have been no swaps (or N swaps) and no mutations then the result is the same genomes. Since this would be a waste of processing, a single gene in each is forced to mutate by selecting a value from 1 to N for each genome and mutating that gene. Unlike with crossover weight, mutation weight is generally kept very low as very high values transform GA into random search.

The last step in the GA is the survival step. First, each genome produced in the current generation is evaluated using the fitness function and assigned a fitness value. Then the genomes are merged into the existing population. Genomes are removed (killed) from the population until it returns to size *P*. Typically, the genomes permitted to survive are those with the highest fitness; however, this can cause particular gene values to become (near) ubiquitous, i.e. less diversity in the population. Therefore, to promote diversity genomes with different gene values may be permitted to survive even if they are less fit.

2.6.4 Particle Swarm Optimization

Like ACS and GA, particle swarm optimization (PSO) is a combinatorial optimization algorithm inspired from nature, in this case from the swarming movements in birds or insects (Eberhart & Kennedy, 1995). PSO consists of a population of size P of particles that are permitted to fly in a hyperspace (hypershape for bounded problems) where the location of the particle in the space corresponds to a solution. This is done by describing the solution space as a N-dimensional hyperspace where each dimension represents an element of the solution and a coordinate represents a specific option for that element. The particles then share information with each on the quality of the solutions found and encouraged to fly towards the more promising areas.

Every particle is described as having a location and a current velocity vector (v_0). Every generation, the particle position is updated based on the velocity vector. Then the velocity vector is modified in accordance with three parameters. The first parameter is inertia (w) which encourages the particle to continue moving in the same direction as v_0. The second and third parameters are the acceleration coefficients (*c1* and *c2*) towards the individual best solution so far (X_{ibest}) and the global best solution (X_{gbest}). The velocity update function is shown in Formula 7, where *rand1* and *rand2* are random values from 0 to 1, and X_{curr} is the current location.

$$v = w \times v_0 + rand1 \times c1 \times (X_{curr} - X_{ibest}) + rand2 \times c2 \times (X_{curr} - X_{gbest}) \qquad (7)$$

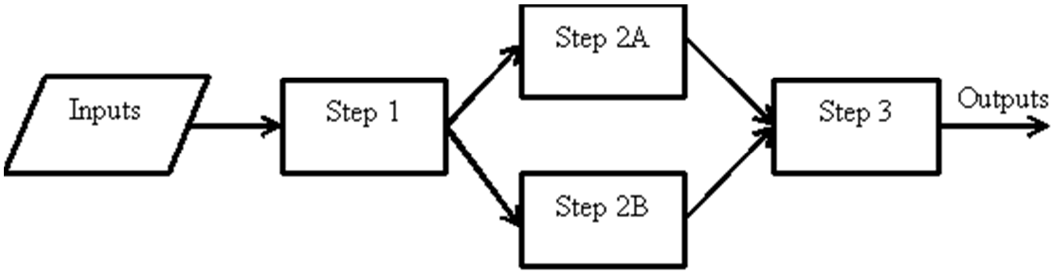
Higher inertia values encourage exploration of the solution as it causes the particle to ignore the best solutions so far. The acceleration coefficients are both exploitation mechanics, with acceleration towards the individual best position being somewhat exploratory as it encourages some local search to see if the individual best solution has a better solution than the current global best. One drawback found in the early research on PSO is particle explosion (Clerc & Kennedy, 2002) where the particles can fly very far from promising areas just from momentum. This is resolved with the maximum velocity (*Vmax*) control parameter which limits the velocity of the particles. For bounded problems, Vmax is never greater than the absolute difference between the lowest and highest bound values (*Xmax*) as a velocity greater than this difference will only cause the particle to strike the hypershape boundary.

2.6.5 Hybrid Architectures

A hybrid CI algorithm is a technique for combining multiple CI algorithms together so as to capitalize on the strengths of each algorithm. Typically, this is done in one of three ways. First, two algorithms may jointly process samples where one is intended to provide a globally oriented search and the other a locally oriented search (Gonçalves, de Magalhães Mendes, & Resende, 2005; Kao & Zahara, 2008). The second technique is to have one algorithm provide a configuration for the second algorithm, e.g. evolving artificial neural networks (Belew, McInerney, & Schraudolph, 1990; Yao, 1999). The third technique is the hybrid architecture where information from one algorithm is sent to other algorithms in the ensemble to improve their processing (Wermter & Sun, 2000).

Hybrid architectures are represented as graph where each node is a CI algorithm and the links represent the data transfer between them. Each algorithm should be considered as steps in an overarching algorithm. Figure 9 shows a very generic example consisting of four algorithms: Step 1, Step 2A, Step 2B and Step 3. Step 1 sends data to both Step 2A and Step 2B, which in turn sends data to Step 3. Each algorithm is trained and executed separately and to completion prior to the next in series, so for example, Step 1 is trained completely first, then Step 2A and 2B and finally Step 3. A loosely coupled hybrid architecture is one where information only moves forward (like in Figure 8); whereas, a tightly coupled hybrid architecture allows for cyclical connections. A tightly coupled hybrid architecture has a shared control and communication mechanism that determines the timings for the training and execution of the individual algorithms.

Figure 8. An example of a loosely coupled hybrid architecture



Hybrid architectures have mainly seen use in language processing and robotics. Jurafsky et al. (1994) combined an ANN and HMM to perform speech recognition. The ANN was trained to identify the phonetics of each word from acoustic features and this identified phonetics were then passed to the HMM for word recognition. Sun and Peterson (1998) combined three algorithms, ANN, a decision making module and decision tree learning to perform robotic navigation. The ANN was trained in an unsupervised learning model to determine the quality of an action (Q-value). The Q-values were then used by the decision making module to choose a course of action. Over a training period the action and corresponding results were used to extract a set of rules which could be provided to a future agent.

For the current research two hybrid algorithms are evaluated. A loosely coupled hybrid architecture is used to improve the precision of identification of learning styles and an evolving artificial neural network (EANN) is used to improve the precision of identification of WMC. For that reason, the EANN is described in greater detail in the following section.

2.6.6 Evolving Artificial Neural Networks

An EANN is a hybrid algorithm which combines an ANN with an evolutionary algorithm to search for an optimal ANN topography (Yao, 1999). Although any evolutionary algorithm may be used, genetic algorithm (Belew et al., 1990; Yao, 1999) and evolutionary programming (Stanley & Miikkulainen, 2002; Yao & Liu, 1997) are typical choices. Since this research uses GA, the remainder of this section will focus on that combination.

One issue with ANNs is the structure of the ANN forces a relationship style between the inputs which may not be optimal. For a simple problem, it might be possible to craft an ANN topology by hand; however, for many non-trivial problems this is not an option as there is insufficient domain knowledge to determine the proper topology. If there were sufficient domain knowledge to derive an optimal function, then the ANN is not necessary. Finding the optimal topology is treated as a combinatorial optimization problem, where the solution space is a bounded area describing a set of possible topologies. The bounds are typically defined as a fixed or minimum / maximum number of hidden layers with a fixed or minimum / maximum number of nodes per hidden layer. For example, the bounds may be to consider all topologies where there is exactly 1 hidden layer and a minimum of 1 node and a maximum of 20 hidden nodes. For the EANN, the solution space is limited to non-cyclical topologies while the recurrent evolving artificial neural network (EANN/R) expands the search to allow for cyclical relationships between nodes.

As discussed above, solving a problem with a GA requires an encoding / decoding scheme for the genome structure. For EANN, there are two encoding / decoding schemes: bit representation and real value representation. For the real value representation, there are two training modes: evolutionary and hybrid. With the bit representation each gene value may be a 0 or 1, and determines if there is a connection between two nodes. For a non-recurrent EANN, Figure 9 shows a sample genome, the decoding of the genome into a matrix and the resulting ANN. For a recurrent ANN, the genomes would have additional genes for the connections marked "n/a" in the matrix and additional genes for connections from the output node (25 genes in total). The real value representation uses a real value from 0 to 1 for each gene which represents the weight of the link.

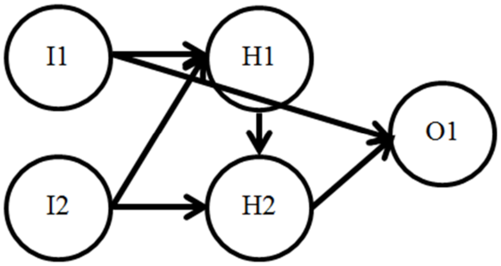
Figure 9. (a) A genome using bit representation (b) Decoded matrix of connections from the genome (c) Corresponding ANN from the matrix

Genes										
0	1	0	1	1	1	0	1	1	1	1

(a)

	I1	I2	H1	H2	O1
I1	n/a	0	1	0	1
I2	n/a	n/a	1	1	0
H1	n/a	n/a	n/a	1	1
H2	n/a	n/a	n/a	n/a	1

(b)



(c)

Two forms of training exist for EANNs: evolutionary and hybrid. Evolutionary training requires the use of a real value representation as the GA is used to find both the optimal topology and optimal weights for the ANN. In other words, the ANN which results from decoding the genome is not altered with any other mechanism, such as backpropagation, and is evaluated as is. With hybrid training, the resulting ANN from decoding the genome is considered a starting point and the ANN is further trained using backpropagation (as in this research) or some other appropriate mechanism. The advantage of hybrid training is that the GA can search for a promising area (global search) and the ANN training method can refine it to the global optimal (local search); whereas, the drawback is an increase in processing time. The processing time needed for hybrid training may be reduced by limiting the number of generations permitted for training the ANN.

Chapter III – Learning Style and Working Memory Capacity Identifiers

This chapter describes how the LSID and WMCID approaches were designed and created to identify learning styles and WMC respectively. This research, like many of the related works (Cha et al., 2006; Chang et al., 2013; García et al., 2007; Graf, Kinshuk et al., 2009; Villaverde et al., 2006), uses student behavior patterns to identify both learning styles and WMC. In order to understand how the CI algorithms were used to identify learning styles and WMC, it is first important to understand the underlying behavior patterns and as such they are described in the first two sections (with learning styles first and WMC second). The third section describes how the CI algorithms were adapted to identify learning styles. This is broken down into sub-sections for the classification algorithm (ANN), optimization algorithms and hybrid algorithms. The last section explains how the CI algorithms were adapted to identify WMC and is similarly broken down into the same three sub-sections as for learning styles. As there is much similarity in how the problems are encoded and how the algorithms were trained for the LSID and WMCID approaches only the differences for WMCID are highlighted in the fourth section.

3.1 Behavior Patterns for Identifying Learning Styles

The behavior patterns used by this research are the same as those used in the development of DeLeS (Graf, Kinshuk et al., 2009) and are shown in Table 5. Most of the behavior patterns relate either to the duration of students' visits to types of learning objects or count how many times a student visits the types of learning objects. Other patterns use the navigational behavior within the LMS as an indicator of learning styles. Lastly, the average grades for different types of questions are considered. For each relevant pattern it is determined whether a high or low data value corresponds to the active, sensing, visual or sequential learning styles and these are marked in Table 5 with a “+“ for a correlation to a high value or “-” for a low value.

Table 5. Relevant behavior patterns for each FSLSM learning style dimension (Graf et al., 2009b)

Active/Reflective	Sensing/Intuitive	Visual/Verbal	Sequential/Global
content_stay (-)	content_stay (-)	content_visit (-)	outline_stay (-)
content_visit (-)	content_visit (-)	forum_post (-)	outline_visit (-)
example_stay (-)	example_stay (+)	forum_stay (-)	question_detail (+)
exercise_stay (+)	example_visit (+)	forum_visit (-)	question_develop (-)
exercise_visit (+)	exercise_visit (+)	question_graphics (+)	question_interpret (-)
forum_post (+)	question_concepts (-)	question_text (-)	question_overview (-)
forum_visit (-)	question_details (+)		navigation_overview_stay (-)
outline_stay (-)	question_develop (-)		navigation_overview_visit (-)
quiz_stay_results (-)	question_facts (+)		navigation_skip (-)
self_assess_stay (-)	quiz_revisions (+)		
self_assess_twice_wrong (+)	quiz_results_stay (+)		
self_assess_visit (+)	self_assess_stay (+)		
	self_assess_visit (+)		

Many of behavior patterns identified by Graf et al. (2009) from literature may be classified into two categories: patterns which pertain to how students visit different types of learning material and patterns which pertain to the relationship between students grades and types of learning material. Graf et al. (2009) also identify four behavior patterns which do not fit into either of these two broad categories. The behavior patterns are summarized below as from their work starting with the patterns which relate to content visitation. This is followed by discussing the patterns which pertain to the relationship between student grades and learning material types, and lastly the description of the unique behavior patterns.

Most of the behavior patterns capture either the number of seconds the student stayed on a particular type of learning material (e.g. *exercise_stay* is the number of seconds the student spent on exercises) or the number of times the student visited the learning material (e.g. *exercise_visit* is the number of times a student visited exercises). The types of learning material are: content, exercises, examples, forum, outline, self-assessment test, course overview (*navigation_overview_stay* and *navigation_overview_visit*) and quiz results (*quiz_stay_results* only, the number of visits to quiz results is not counted).

The average grade behavior patterns (prefixed with *question_*) are divided into six categories concerning what type of material is covered by the question: concept, details, development (develop), facts, interpretation (interpret) and overview. Additionally, there are two question types for how the related learning material is delivered: textually (text) or graphics. A question may have more than one type, for example, a question on text-based factual learning material would belong to both the *question_facts* and *question_text* behavior patterns.

The remaining patterns are unique and described as follows. The *forums_post* behavior pattern counts the number of postings the student made to the course forums. The *navigation_skip* pattern counts the number of times that a student skips over learning material. The *quiz_revisions* pattern counts the number of times the student altered an answer to a quiz prior to submitting it. Lastly, *self_assess_twice_wrong* counts the number of times a student is incorrect on the same self-assessment test question.

As seen in Table 5, not all of the behavior patterns are relevant for each of the FSLSM dimensions. The following four sub-sections relate the behavior patterns to the characteristics of a learning style dimension. All of the descriptions below are based on the work of Graf (2007).

3.1.1 Active / Reflective Dimension Behavior Patterns

Students with an active preference tend to be more interested in using material than reflecting about it. Thus, these students are expected to visit with more frequency exercises and self-assessment tests and stay longer on exercises. Inversely, reflective students prefer to think about material and so are expected to visit more frequently and stay longer on content. Although both active and reflective students are expected to visit examples with the same frequency, reflective students are more likely to stay longer than active students both because reflective students will prefer to think about the example and active students prefer to try solving a problem for themselves in the exercises. Active students tend to prefer communicating with others, while reflective students prefer to read and reflect on what others say. Thus, active students are expected to post to forums with greater frequency while reflective students will visit forums more often to read what has been posted. Reflective students will tend to stay longer on outlines, quiz results and self-assessment results and since they tend to think longer on their answers for self-assessment tests they are less likely to get them wrong twice (i.e. getting a self-assessment question wrong twice is an indication of an active preference).

3.1.2 Sensing / Intuitive Dimension Behavior Patterns

Students with a sensing preference tend to prefer facts and details, which may then be related to the real world (i.e. the world of sensory experience). Sensing students tend to be practical realists. Intuitive students prefer to learn abstract concepts and principles. Intuitive students tend to use this conceptual knowledge in more creative ways. Thus intuitive students prefer conceptual material and so will visit content more frequently and for longer periods of time. Intuitive students will tend to do better on questions about conceptual material as they will prefer to study it and they will tend to do better on questions which require them to develop a new solution from concepts (*question_develop*). Sensing students prefer concrete and procedural material and so will prefer reinforcing this knowledge from examples. Similarly, they will prefer to do exercises and self-assessment tests since it allows them to put the procedures they've learnt into practice. Sensing students tend to work more carefully than intuitive students and so it is expected that they will revise quiz answers more often before submitting them. Similarly, sensing students will prefer to review quiz results more often. Sensing students will tend to do better on questions which require details or are concerned with factual learning material.

3.1.3 Visual / Verbal Dimension Behavior Patterns

Since most content requires reading, verbal students will have a greater tendency to visit learning material in general. Verbal students will also prefer to read and write to course forums, thus visiting the forums more often, staying for longer periods of time and having more posts. Verbal students will tend to do better on questions on learning material which was provided textually; whereas, visual students will do better on questions from learning material that was presented graphically.

3.1.4 Sequential / Global Dimension Behavior Patterns

Sequential students prefer to navigate / learn in a linear fashion, so these students are less likely to skip learning material. Global students will be more likely skip ahead to more advanced learning material as they prefer to understand how it all fits together. Global students are also more likely to visit more frequently and stay for a longer period on content outlines and course overviews, as these often provide a look at how the learning material fits together. Since sequential students are more interested in understanding each piece of content in depth before moving on to the next in sequence, they are more likely to do well in questions that focus on the details of the material. Global students are more likely to do well on questions that focus on a broad overview of material. In addition, global students are more likely to do well on questions which require interpretation or developing a new solution, as this generally requires a broader understanding of the material and the relationships between concepts with the material.

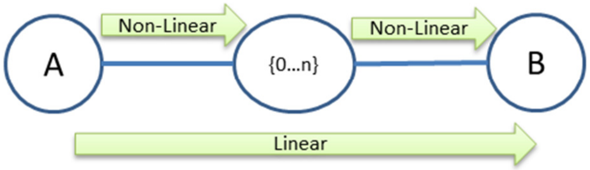
3.2 Behavior Patterns for Identifying Working Memory Capacity

The patterns used to identify WMC are those proposed by Chang et al. (2013) from their review of psychology literature consisting of five navigational patterns exhibited by students when using a learning system and three of the four learning style dimensions as described by the FSLSM. Each of the navigational patterns has an activated (*act*) and non-activated (*nonact*) state where activated means that the particular pattern has been exhibited by the student and non-activated that it has not. In what follows there is a description of each indicator and then a discussion on how these indicators are used to identify WMC. The descriptions of the behavior patterns are based on the work by Chang et al. (2013).

3.2.1 Linear Navigation Pattern

The first pattern is the *linear navigation pattern* which is described as when a student progresses through learning objects (LOs) as intended by the teacher or course designer. The basis of this pattern is the work of Huai (2000) which investigated the relationship of linear and non-linear navigational behaviors and WMC. Huai found that students who exhibited linear navigation tended to have high WMC and vice versa. Figure 10 shows an example of linear and non-linear navigation. When exhibiting a linear navigational pattern the student goes directly from learning object A to learning object B; whereas, the non-linear navigational pattern has the student visiting other LOs in-between A and B.

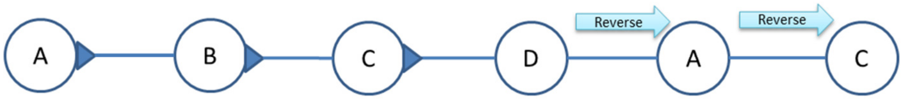
Figure 10. An example of linear navigation (Chang et al., 2013)



3.2.2 Constant Reverse Navigation Pattern

The second pattern, *constant reverse navigation pattern*, is defined as navigating two or more times to a previously visited learning object in an order not intended by the teacher or course designer. A student exhibiting such a pattern may be unable to recall recently visited LOs which may indicate they have low WMC (Lin, Kinshuk, & Patel, 2003). Figure 11 shows an example of this pattern. In this example, the student navigates as intended by the course design from learning object A to B to C to D. The student then revisits A from D and then from A to C both of which are not intended by the course design. It is at the point of revisiting learning object C, the second revisit, that the constant reverse pattern would be considered activated.

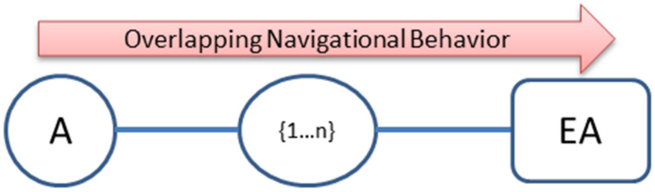
Figure 11. Example of constant reverse navigation pattern (Chang et al., 2013)



3.2.3 Simultaneous Tasks Pattern

The *simultaneous tasks pattern* is based on psychological studies showing that when a student with low WMC attempts to perform simultaneous tasks they will have an increase in errors (Engle, 2010; Woehrle & Magliano, 2012). For this pattern, a student is considered to be performing simultaneous tasks if there is an overlapping navigational behaviour. This is defined as visiting a learning object (LO A) and then visiting one or more other learning objects prior to doing the evaluation of LO A (EA) as shown in Figure 12. If the student passes the evaluation of LO A then this indicates a high WMC as they were able to recall information even with intervening activity, while failing the evaluation indicates low WMC.

Figure 12. Example of overlapping navigational behavior (Chang et al., 2013)

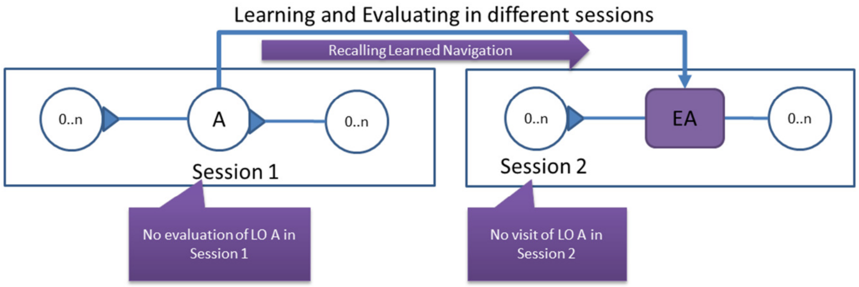


3.2.4 Recalling Learned Material Pattern

The *recalling learned material pattern* is very similar to the simultaneous tasks pattern above except that the visitation of the learning object and corresponding evaluation are done in different learning sessions. It is detected when the student completes a learning object A (LO A) and then completes the evaluation for LO A (EA) in a subsequent learning session. Additionally, the student may or may not visit additional learning objects in either of the two sessions prior to completing EA. As with the simultaneous tasks pattern, if the student is able to recall sufficient information to pass EA, then this indicates a high WMC, while if they fail EA this indicates a low

WMC. An example of this navigational pattern is shown in Figure 13. This pattern is based on works showing that a student's ability to retrieve information from long term memory is related to their WMC (Engle, 2010; Unsworth, Redick, Spillers, & Brewer, 2012)

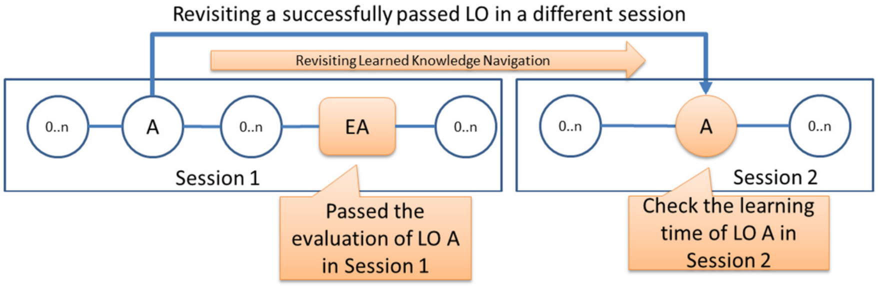
Figure 13. An example of learning and evaluating in different sessions (Chang et al., 2013)


Learning and Evaluating in different sessions
Recalling Learned Navigation
0..n
A
0..n
Session 1
0..n
EA
0..n
Session 2
No evaluation of LO A in Session 1
No visit of LO A in Session 2

3.2.5 Revisiting Passed Learning Object Pattern

Based on the same concept as the recalling learned material pattern, that a student's ability to recall information from long term memory is related to their WMC (Engle, 2010; Unsworth et al., 2012), this pattern seeks to identify when a student is not able to recall information from long term memory. If a student has visited learning object A (LO A) and then passes the evaluation of this learning object (EA), it is expected that they should know the material. Thus, if the student then revisits LO A in a future session, it indicates a difficulty in recalling the information from long term memory and may indicate low WMC. An example of this navigational pattern is shown in Figure 14. To determine if this pattern is activated for student, the time spent in each LO is recorded as a base line for each student (b_i). If a student visits the LO again the time spent relearning the material is recorded (v_i). The ratio (r_i) between the time spent relearning versus the initial base line for the student is then calculated as shown in Formula 8. For each learning object, an average of ratio values (r_{avg}) is calculated as shown in Formula 9. If a student's $r_i > r_{avg}$ then the student has taken longer than average to learn and relearn the material and this indicates a low WMC. High WMC is indicated by a student's $r_i \leq r_{avg}$.

Figure 14. An example of relearning in a subsequent session (Chang et al., 2013)


Revisiting a successfully passed LO in a different session
Revisiting Learned Knowledge Navigation
0..n
A
0..n
EA
0..n
Session 1
0..n
A
0..n
Session 2
Passed the evaluation of LO A in Session 1
Check the learning time of LO A in Session 2

$$r_i = \frac{v_i}{b_i} \tag{8}$$

$$r_{avg} = \frac{\sum_{i=1}^{n} r_i}{n} \tag{9}$$

3.2.6 Learning Styles as Indication of WMC

Graf et al. (2007) investigated the relationship between FSLSM learning style dimension and cognitive traits. They conclude that with respect to WMC there is a relationship for the A/R, S/I and V/V dimensions. The study found that students with a reflective or intuitive learning style tend to have a high WMC, while those with an active or sensing learning tend to have a low WMC. Although verbal students were found to have a high WMC, visual students were found to have either high or low WMC. In DeWMC (Chang et al., 2013), three indication values ($i_{a/r}$, $i_{s/i}$, $i_{v/v}$) are computed from the strength of the student's A/R, S/I and V/V preferences. The $i_{a/r}$ and $i_{s/i}$ are real values from

0.0 to 1.0 and are proportional to the strength of student's A/R and S/I preferences with a 0 assigned for a maximal active and sensing strength and 1 for maximal reflective and intuitive strength. The $i_{v/v}$ indicator ranges can be unassigned or range from 0.5 to 1.0. If the student has a visual preference it is left unassigned; however, for students with a verbal preference the value is proportional to the strength of the preference with 1.0 for the highest verbal preference. The overall WMC hint is then found by averaging the two or three indication values (two if $i_{v/v}$ is unassigned).

3.3 Learning Style Identifiers

The aim of this section is to detail how the five CI algorithms are used to identify learning styles through the development of the LSID approaches. In developing the LSID approaches, we sought two key qualities: generic and high precision. By being generic the LSID approaches may be integrated into any LMS instead of being tied to a specific system. Higher precision means that students will be provided with more accurate recommendations and personalization. Genericity is achieved by using behavior patterns which are general to any LMS, while precision is measured by performance metrics described later in the next chapter. In addition to these two qualities, the LSID approaches were evaluated (using performance metrics) on how fair they were to students. Fairness means that all students should be measured with approximately the same precision, as individual students who are not precisely identified may suffer as a result through no fault of their own. The remainder of this section will discuss the directions investigated to design and develop the LSID approaches. This is followed by a sub-section on each broad approach used by this research: classification, optimization and hybrid algorithms.

The first step was to investigate how CI algorithms could be used to improve the precision of learning styles identification. From reviewing the literature, it was decided that a novel approach would be to take an existing leading approach and improve on the precision by using CI algorithms. This has the advantage of knowing that the approach is already somewhat precise and therefore makes for a good basis. DeLeS (Graf et al., 2009b) was selected for this purpose as it had the best overall results of the approaches found in literature and used behavior patterns generic to any LMS. The second step was to decide on how to improve precision using CI algorithms and capitalizing on DeLeS.

It was decided to do the second step using three types of algorithms but in two phases. In the first phase, two types of algorithms were used: a classification algorithm and optimization algorithms. The classification algorithm, the ANN, uses the behavior patterns identified for designing DeLeS (Graf, Kinshuk et al., 2009) as inputs by reason that an ANN would be able to find a more precise function for identifying learning styles. The reasoning for the optimization approach was that DeLeS was already effective at identifying learning styles but uses an unweighted average in its calculation. Another way of looking at DeLeS is it used a weight of 1.0 for each behavior pattern, and this set of weights is unlikely to be optimal. An optimization was used to optimize the pattern weights and thereby improve the precision of identification. The second phase of step two analyzed the results from the first phase and selected a hybrid algorithm to overcome the weakness of the mono-CI algorithms and to further improve results. Based on the analysis, a loosely couple hybrid architecture was selected. The design of each approach, i.e. how each algorithm was adapted to identify learning styles, is described in the following sub-sections.

3.3.1 Improving Learning Styles Identification through Classification

There are many different classification algorithms; however, for this research the ANN was selected for three main reasons. Given the size of the training set it was reasoned that statistical classifiers would not work as well (Niles, Silverman, Tajchman, & Bush, 1989). Other non-statistical classifiers have been well researched and found to be not as successful as rule-based approaches (Carmona et al., 2008; Cha et al., 2006; García et al., 2007; Özpolat & Akar, 2009). Although there does exist an ANN based approach in literature (Villaverde et al., 2006), our approach differs in three key manners. First, this research identified each learning style with a separate ANN, i.e. we use four different ANNs instead of one. Second, LSID-ANN used more granular real valued data as inputs instead of integer data. Third, the outputs from LSID-ANN were a real value from 0 to 1 as opposed to only a 0 and 1. This means that LSID-ANN was able to identify the strength of the student's preference.

As discussed previously, ANNs consist of a graph with weighted links and by adjusting the weights on the links very complex functions can be represented. As such ANNs are have been called the universal approximator (Hornik, Stinchcombe, & White, 1990) and could be successful at finding an efficient function for identifying learning styles.

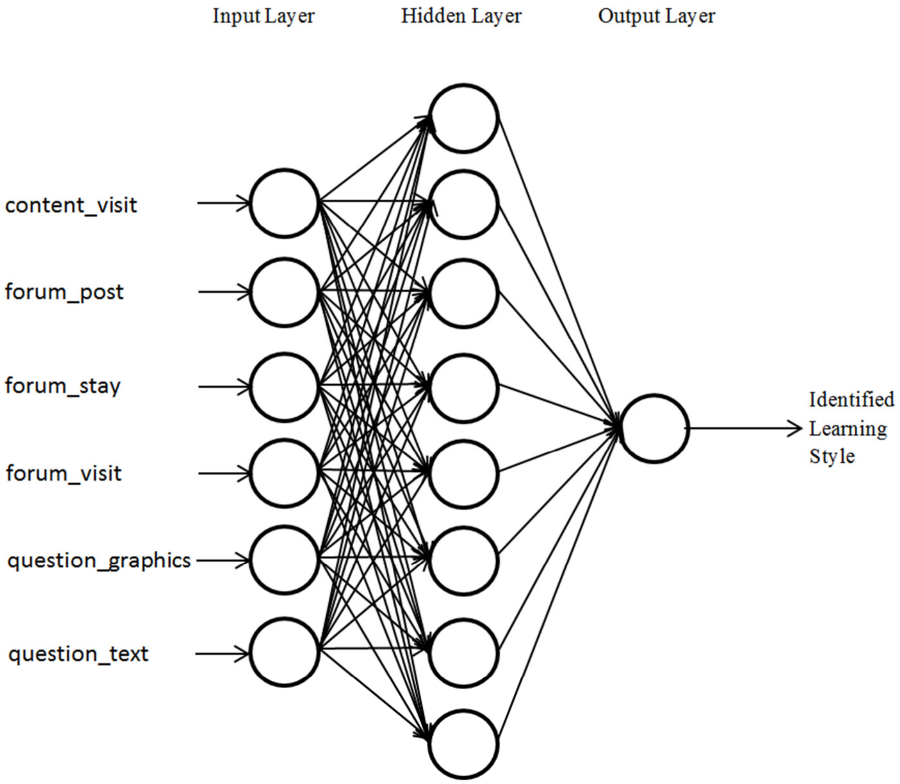
The inputs to each ANN are the relevant behavior patterns for the learning style dimension under consideration. So, when LSID-ANN is built to identify the A/R dimension it has 12 inputs, while for the S/I dimension it has 13 inputs, 6 inputs for the V/V dimension and 9 inputs for the S/G dimension. Initially, the inputs for LSID-ANN were used without pre-processing; however, it was observed that the resulting learning styles would have very large changes from generation to generation and the resulting precision was worse than DeLeS. This effect from using inputs of different scales is discussed in literature (Priddy & Keller, 2005), although it does not always occur for all problems; when it does occur it is recommended to normalize the data (Priddy & Keller, 2005). Initially, the values were normalized to the minimum and maximum of each behavior pattern; however, since most behavior patterns contain an outlying data point the normalized values were very similar and quite small. The results with this normalization were once again worse than DeLeS. The second, and final, technique normalized to the upper threshold value, which is the boundary between balanced and a strong preference, (T_{up}) for each behavior pattern resulting in Formula 10 and this normalization formula was used for optimizing parameters, evaluating overfitting reduction and producing the final result. For LSID-ANN, regardless of learning style dimension, there is always a single output which produces a real value from 0 to 1 which is taken as the identified learning style value. The number of hidden nodes is a control parameter and was optimized for each learning style dimension (this process is described in section 3.8.1). A sample of the topography used for the V/V dimension is shown in Figure 15.

$$V' = \frac{V}{T_{up}} \tag{10}$$

Since the training data contains the actual student learning styles a supervised learning model may be used. Backpropagation is used to train an ANN with a supervised learning model which requires an error calculation. The error (e) for LSID-ANN is calculated as the difference between the actual and identified learning style values, with a precision as 1-e. With individual training, after each student, the ANN weights are adjusted to reduce the error; otherwise the weight adjustments are recorded, summed and applied at the end of the generation. After each generation, the fitness function is used to assess the overall quality of the current topography as the average of the precision values across all students. This process continues until the termination condition is reached.

For this research, the termination condition is picked to promote finding the optimal solution by using three rules. The first rule states that whenever a new best result, as calculated by the fitness function, occurs the current generation is recorded (G_{best}). The second rule states if G_{best} number of generations have passed since finding the last best result then stop processing. The third rule states that a minimum (10,000) generations must first pass before terminating. This termination condition is also used for all other LSID and WMCID approaches.

Figure 15. LSID-ANN topography for the V/V dimension



3.3.2 Improving Learning Styles Identification through Optimization

This sub-section describes how optimization algorithms were adapted to improve the precision of learning styles identification. For this research, three optimization algorithms, ant colony system, genetic algorithm and particle swarm optimization were selected to build three corresponding approaches LSID-ACS, LSID-GA and LSID-PSO. Each of these three approaches is based on DeLeS (Graf, Kinshuk et al., 2009). DeLeS works by averaging a set of hint values (*h*) calculated based on the relationships between students' behaviors when using learning systems and learning styles. Since there are few, if any studies, on the importance of each behavior pattern towards identifying learning styles, DeLeS assumes that the weight (*W*) of each behavior pattern is 1.0. The weighted average calculation used is shown in Formula 11. The three LSID approaches listed above aim to address this limitation by searching for an optimal weight for each behavior pattern. The solution space for this problem tends to be rather large with 10^{12} to 10^{26} combinations depending on the learning style dimension. Although brute force searching guarantees that an optimal solution will be found such algorithms become intractable with larger solution spaces (Russell & Norvig, 2010). Thus, more efficient searching algorithms were needed to effectively solve this problem. In reviewing literature, it was found that ACS, GA and PSO have been effective for finding optimal weights and so selected for this research (Abido, 2002; Ericsson, Resende, & Pardalos, 2002; Pothiya et al., 2010). In the next paragraphs, DeLeS is introduced in more detail in order to provide background information on how the proposed LSID approaches work. Subsequently, a general overview is provided on how the proposed LSID approaches work and after that, each algorithm is described separately in a designated subsection.

$$LS = \frac{\sum_{x=1}^{n} W_x \times h_x}{\sum_{x=1}^{n} W_x} \tag{11}$$

To identify learning styles, DeLeS extracts behavior pattern data from the learning system's database and translates the behavior pattern data into a learning style hint h. A hint value of 3 indicates that the behavior pattern data provide a strong indication for an active, sensing, visual or sequential learning style (according to Table 5. A hint value of 2 indicates that the student's behaviour is average and therefore does not provide a specific hint towards a learning style. A hint value of 1 indicates that the behavior pattern data provide a strong indication for a reflective, intuitive, verbal or global learning style (according to Table 5). A hint value of 0 indicates that no information about the student's behavior is available with respect to the respective pattern. In order to classify the behavior pattern data into learning style hints, an upper and lower threshold has been derived from literature and is used for each pattern. To calculate the learning style of a student in a given learning style dimension, an average hint value is calculated by summing up all hint values of all patterns relevant for that dimension and dividing it by the number of patterns that include available information (for the respective dimension). The resulting value is then normalized, leading to a value between 0 and 1, where 1 indicates a strong preference for an active, sensing, visual or sequential learning style and 0 indicates a strong preference for a reflective, intuitive, verbal or global learning style.

Each LSID approach follows the same overall process to calculate learning styles as DeLeS, with only one difference. When DeLeS is calculating learning styles from hint values, it calculates an average hint value where each hint from a behavior pattern contributes equally to the overall learning style value. The LSID approaches use different optimization algorithms to identify optimal weights for each behavior pattern and therefore, when calculating learning styles from hint values, a weighted average hint value is computed (instead of just an average hint value), where each hint value is multiplied by the optimal weight of the respective pattern.

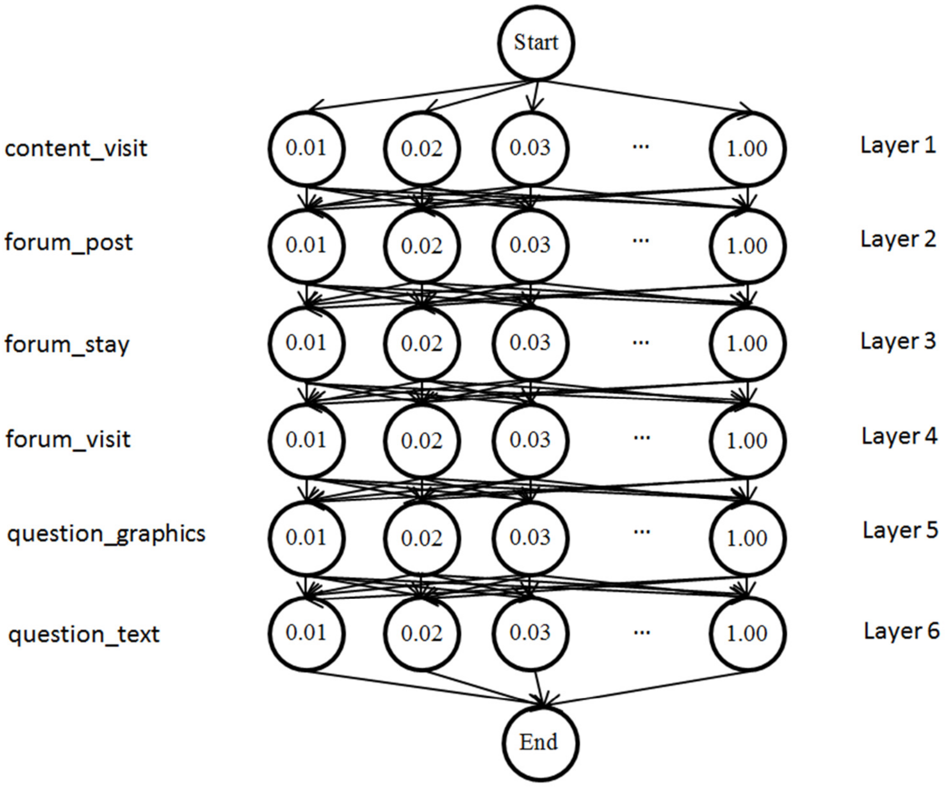
For each of the optimization algorithms, the same termination condition is used as previously described for LSID-ANN. The fitness function for the optimization algorithms works by subtracting the error between the actual (LS_{actual}) and identified ($LS_{identified}$) learning styles from 1. This is done for each student in the training set of size T and then averaged to give the overall fitness (shown in Formula 12).

$$F = \frac{\sum_{x=1}^{T}(1 - |LS_{actual,x} - LS_{id,x}|)}{T} \tag{12}$$

3.3.2.1 Ant Colony System – LSID-ACS

ACS requires the problem to be converted into a graph for the ants to traverse. In this case, the problem is to find a set of optimal weights for the behavior patterns related to each learning styles dimension. To represent this problem, a layered graph is built beginning with a start node simply to make for a convenient entry point. Then, the start node is connected to each node in a layer of 100 nodes with values 0.01 to 1.0 in increments of 0.01 called “Layer 1”. This layer is then repeated N-1 times, where N is the number of behavior patterns. Each node in “Layer X” is connected to each node in “Layer X+1”. “Layer 1” represents the possible weights for the 1st behavior pattern, “Layer 2” for the 2nd behavior pattern and so on. Lastly, for the convenience of knowing when an ant is finished traversing the graph, every node in “Layer N” is connected to an exit node. When an ant is finished traversing the graph it will have selected a single node from each layer, and this forms the candidate set of weights. The graph for the V/V dimension showing how each layer corresponds to a behavior pattern's possible weights in shown in Figure 16.

Figure 16. LSID-ACS' graph for finding set of weights for V/V dimension



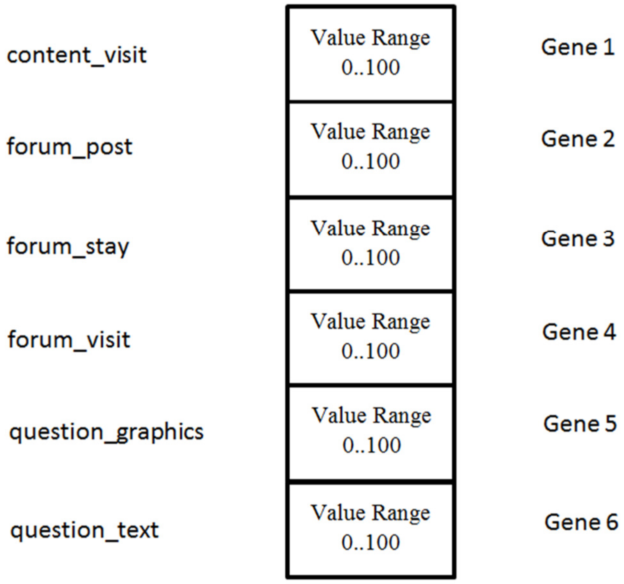
After the graph is built, the local quality of the graph's links are populated; however, since there is no initial information on what weights might be good choices each of the values are set to 1.0. Furthermore, with no initial information on the potential quality of weight values no candidate lists are constructed and the ants use only the pseudorandom proportional rule. Since the graph is unidirectional, the ants cannot return to a previously selected node during a single pass, so no tabu lists are used.

After the graph has been constructed and initialized, the population of ants is built. The following process is then followed for each ant in each generation until the termination condition is reached which is identical to that described for LSID-ANN. Each ant is placed in the “Start” node and permitted to traverse the graph using the pseudorandom proportional rule until it reaches the “End” node. When an ant traverses a link it consumes a portion of the pheromone in proportion to the consumption ratio. The path from each ant is decoded into a set of optimal weights and assessed using the fitness function which is the average of the precision values across all students as described for LSID-ANN. Once all of the ants’ fitness values have been calculated, if the best ant has a fitness value greater than the current global best then its path is saved. The links along the global best path have their pheromone values updated in proportion to its fitness value. Finally, all the links in the graph lose a proportion of pheromone in proportion to the evaporation ratio.

3.3.2.2 Genetic Algorithm – LSID-GA

To find the optimal pattern weights, the genome structure uses N gene values, where N is the number of relevant behavior patterns for the learning style dimension. Each gene is permitted to have an integer value from 1 to 100 representing the weight for a behavior pattern. The first gene is the weight for the first behavior pattern while the second gene is the weight for the second behavior pattern, and so on. Thus, the genome as a whole provides a set of weights as a candidate solution. The genome structure shown in Figure 17 shows how each gene represents a weight range for the behavior patterns in the V/V dimension.

Figure 17. LSID-GA's genome structure for the V/V dimension



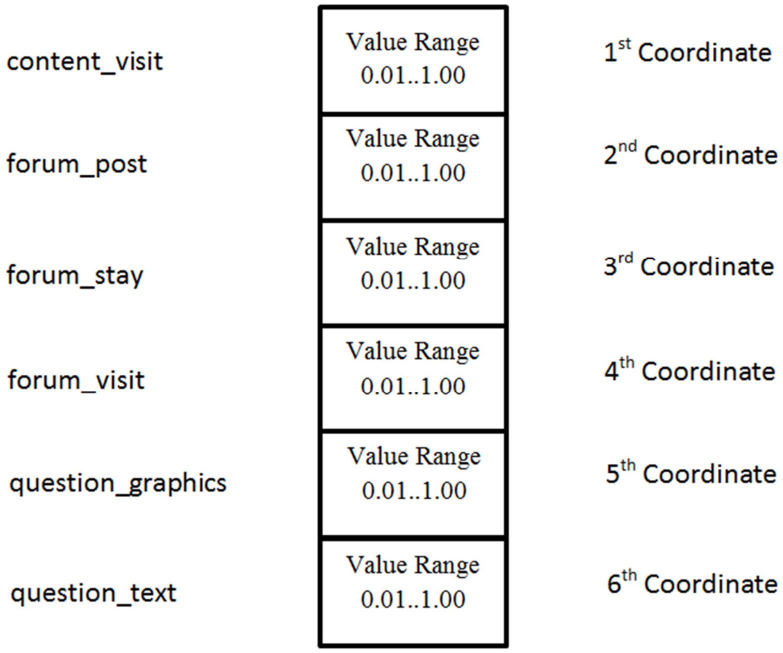
LSID-GA uses simple and well-known GA operators. During initialization, the population is fully populated with *P* genomes using random gene values as no initial information is available on the potential quality of any weight value. For the selection operator, the roulette wheel technique is used where the odds of any genome being selected is equal to its fitness divided by the total fitness of all genomes in the population. The selection operator picks P/2 genomes pairs. Although a genome may be selected more than once the same pairing may not be selected in a single generation so when this occurs a new pair is selected. The crossover operation uses uniform crossover on each genome pair to produce new offspring where each gene has a chance of being swapped equal to the crossover weight. The mutation operator is applied to each of the new offspring where each gene has a chance to be mutated equal to the mutation weight. There is a small chance that no crossover will occur, depending on the crossover weight. When this occurs, the resulting offspring would be identical to the parents and so to ensure some difference one gene is forced to mutate. Following mutation, the genomes are assigned a fitness value using the fitness function which operates as described for LSID-ANN. The new genomes are then merged into the population and an elitist survival strategy is used culling the genomes with the lowest fitness until the population reaches size P. This process is repeated for each generation until the termination condition is reached which is identical to that described for LSID-ANN.

3.3.2.3 Particle Swarm Optimization – LSID-PSO

PSO requires a hyperspace or hypershape for the particles to fly in. For this research, an N-dimensional hypercube is defined, where N is the number of behavior patterns for the learning style dimension under consideration. Each hypershape dimension represents the range of possible weights for a behavior pattern and so each hypershape dimension is bounded with a minimum value of 0.01 and a maximum value of 1.0. The coordinate space is shown in Figure 18 with the coordinates decoded such that the first coordinate is the weight for the first behavior pattern; the second coordinate is the weight for the second behavior pattern, and so on.

First, a population of particles are created with randomized positions and initial velocities. The following process is then followed for each generation until the termination condition is reached which is identical to that described for LSID-ANN. The particles are moved in accordance with the individual velocity. The position of each particle is then decoded into a candidate solution and the particle is assigned a fitness value as the precision values across all students as previously described for LSID-ANN. If the fitness value for a particle is greater than its individual best fitness value, then the position is saved as its individual best. If the highest fitness value for any particle is greater than the global best fitness value found so far, then that particle’s position is recorded as the global best position. The velocity vector of each particle is then modified in accordance with the algorithm's parameters (inertia and acceleration coefficients), the individual and global best positions and two random real values from 0 to 1 (as described in section 2.6.4).

Figure 18. LSID-PSO's coordinate to behavior pattern for the V/V learning style dimension



content_visit	Value Range 0.01..1.00	1st Coordinate
forum_post	Value Range 0.01..1.00	2nd Coordinate
forum_stay	Value Range 0.01..1.00	3rd Coordinate
forum_visit	Value Range 0.01..1.00	4th Coordinate
question_graphics	Value Range 0.01..1.00	5th Coordinate
question_text	Value Range 0.01..1.00	6th Coordinate

3.3.3 Improving Learning Styles Identification through Hybrid Algorithms

The motivation to use a hybrid algorithm is that hybrids may compensate for the weakness of mono-CI algorithms and capitalize on their strengths. So the first step in selecting and designing a hybrid algorithm was to examine the results from the mono-CI algorithms. The key observation made was that multiple executions of the mono-CI algorithm based LSID approaches (e.g. LSID-ANN) resulted in similar average precision but different results on a student by student basis. As a sample of this behavior, Table 6 shows the results from two executions of LSID-ANN, called LSID-ANN-1 and LSID-ANN-2, for the A/R dimension. It is observed that students A and B are identified more precisely in the first execution while students D, F, G and H are identified best in the second execution. This suggested that the problem of learning styles identification is not optimally solved by a single algorithm producing a single solution. This approach improved the precision of learning styles identification by splitting the data set into optimal subsets which are then identified by an algorithm specialized for their respective subset. To continue this example, it would be as if the data were to be divided into the groups G1=(A,B,C,E) and G2=(D,F,G,H) (note students C and E could be in either group as the result for this student is the same) and then the students in G1 are identified by LSID-ANN-1 and the students in G2 are identified by LSID-ANN-2 and so resulting in the “Optimal” row (shown in Table 6).

Table 6. Precision results from two executions of LSID-ANN (best result bolded)

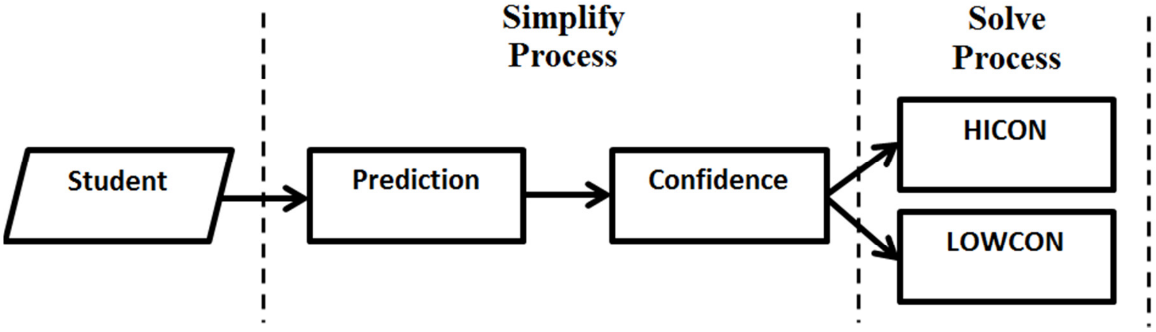
	Precision								
	A	B	C	D	E	F	G	H	Avg
LSID-ANN-1	**0.72**	**0.85**	**0.92**	0.95	**0.80**	0.81	0.83	0.67	0.82
LSID-ANN-2	0.51	0.67	**0.92**	**0.97**	**0.80**	**0.96**	**0.93**	**0.79**	0.82
Optimal	0.72	0.85	0.92	0.97	0.80	0.96	0.93	0.79	0.87

Speaking broadly, a *simplify process* divides the student data set and sends each student to an appropriate algorithm in the *solve process* which does the actual identification. By splitting the data it is reasoned that each algorithm in the solve process need only solve a subset of the students, and this should be a simpler problem (hence the algorithm’s name “simplify and solve” or SISO). Furthermore, since each solving algorithm is concerned only with a subset of students, they should specialize and hence further improve their individual solutions to be more precise for their subset of data than if they trained with the entire dataset. To continue the example from before, it is reasoned that if LSID-ANN-1 need only identify students A,B,C and E it should be able to find a better solution for these four students than it found for them when it was trying to identify all eight students.

The first step in designing the algorithm was to determine the process for spliting the student data. The first process considered, perhaps as it was the most intuitive, was to use the output from an LSID approach (e.g. LSID-ACS) to do an initial identification of the students, with students identified with a high preference (≥ 0.5) sent to one solving algorithm and the rest to another. This proved unsuccessful (identical results to the mono-CI approaches) and on further reflection this makes sense. In effect, splitting the dataset this way is roughly equivalent to the original problem: using a mono-CI to find a single solution to positively identify learning styles for an entire data set. The second process investigated was to use split the data set based on whether the initial prediction is likely to be correct or not. To do this, a classification algorithm is trained to classify, based on the behavior pattern data, the initial prediction into two categories: high confidence and low confidence, i.e. those which are likely to be correct and those which are not. Those students with an initial prediction that has a high confidence are sent to a CI algorithm called HICON for a final identification. Similarly, those students whose initial prediction has low confidence are sent to a separate CI algorithm called LOWCON for re-identification. It may seem counter-intuitive to identify students whose initial prediction has high confidence since these are assumed to be essentially correct; however, since the data split is not perfect some misidentified students are sent to HICON, and this gives these students a chance to be re-identified properly.

This second process described above is then developed as a loosely coupled hybrid architecture (Wermter & Sun, 2000) forming an approach called "Learning Style Identifier – Simply and Solve" (LSID-SISO). The architecture starts (shown in Figure 19) with a student behavior pattern data extracted from the learning system. The *Simplify Process* consists of the prediction and confidence steps described above. The *Solve Process* consists of the HICON and LOWCON steps. Each of these steps consists of a single CI algorithm, so there are four CI algorithms in total within the LSIS-SISO architecture.

Figure 19. LSIS-SISO architecture



The first step is the *Prediction* step (Figure 19) which uses an LSID approach to make an initial prediction of a student's learning styles. The analysis of the mono-CI algorithm-based approaches showed that ACS and ANN each are best in precision for two of four learning style dimensions. So both ACS and ANN are evaluated to find which performs best for each learning style dimension when used as part of the LSID-SISO approach. The initial prediction with ACS operates exactly as described previously for LSID-ACS, and similarly the ANN operates exactly as LSID-ANN. As with the other LSID approaches, each learning style dimension is treated as a separate problem and so eight LSID-SISO algorithms are developed and evaluated. Four of these algorithms use ACS for the prediction step and are called LSID-SISO (ACS). The remaining four algorithms use ANN for prediction step and are called LSID-SISO (ANN).

For the *Confidence* step (Figure 19), an algorithm virtually identical to LSID-

ANN is used. An ANN was selected as, on examining the results from the mono-CI algorithm approaches, it was found that the ANN was the most consistent across all dimensions. Since the inputs and problem are similar to identifying learning styles, it is reasoned that it should do well. The ANN is similar to LSID-ANN having only a few differences. The Confidence ANN uses the initial predicted value as an additional input, with no pre-processing of the value needed as it already ranges from 0 to 1. The output of the Confidence ANN is not decoded as a learning style value but as a belief that the initial prediction should be regarded with low or high confidence. High confidence is defined as having a threshold value ≥ 0.75 (T_{conf}) and was found experimentally by trying values from 0.50 to 0.95 in increments of 0.05. For the purposes of training, a supervised learning model can be used for the *Confidence* step as an expected value can be computed. Since the *Prediction* step is fully completed before starting the *Confidence* step an initial predicted learning style ($LS_{predicted}$) is known. Similarly, the actual learning style (LS_{actual}) is known from the student data. From this an expected confidence value can be determined. If $\mathrm{LS_{predicted}} - \mathrm{LS_{actual}} \leq 1 - \mathrm{T_{conf,}}$ then the initial prediction is considered to be of decent quality and therefore an expected value of 1 is used (high confidence); otherwise, the prediction is considered inaccurate and an expected value of 0 is used (low confidence). An error (*e*) from the expected value and the actual output can then be calculated and used as the fitness value. The termination condition for the Confidence ANN is identical to that previously described for LSID-ANN.

For the *Solve* step (Figure 19), HICON and LOWCON use ANNs for the same reason as above, LSID-ANN has the best overall performance of the mono-CI algorithm approaches when considering all of the results for all learning style dimensions. Both the

HICON and LOWCON are identical to LSID-ANN except for two additional inputs: the initial predicted value and the confidence value. The output is decoded as the identified learning style value as for LSID-ANN. Figures 20 and 21 show the LSID-SISO (ACS) and LSID-SISO (ANN) architectures from the point of view of the algorithms used for each step.

Figure 20. LSID-SISO (ACS) Architecture

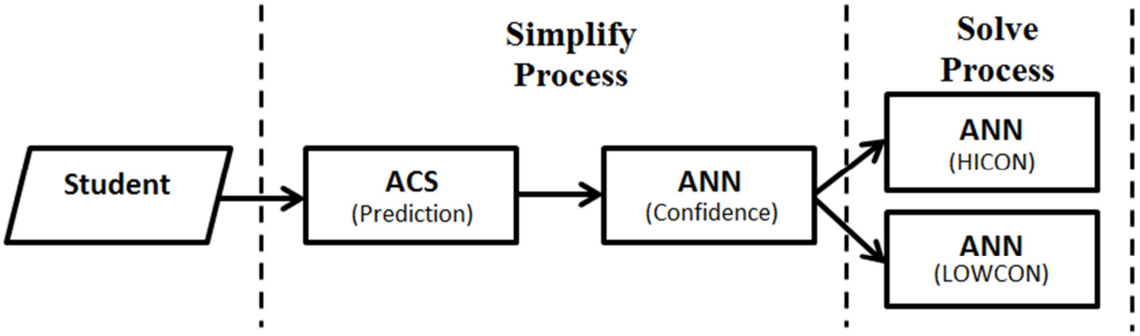
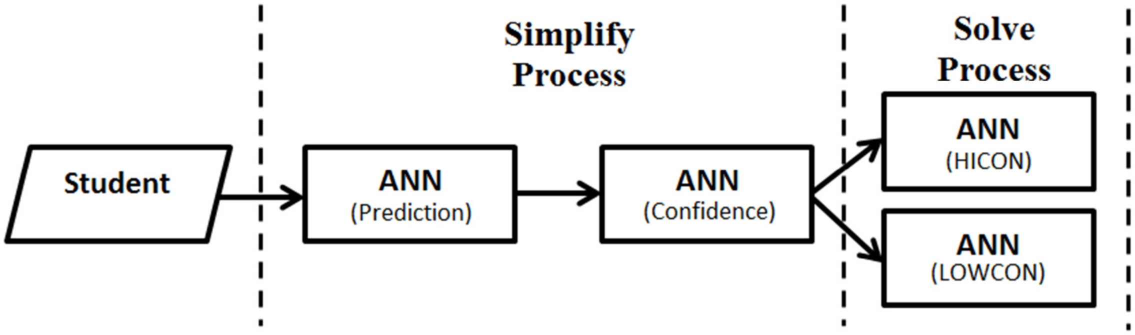


Figure 21. LSID-SISO (ANN) Architecture



3.4 Working Memory Capacity Identifiers

Similar to the previous section, the aim of this section is to discuss how the five CI algorithms are used to identify WMC through the development of the WMCID approaches. The same two key qualities (generic and high precision) were sought for the WMCID approaches for identical reasons and also the WMCID approaches were evaluated for fairness. As with learning styles identification, genericity is achieved by using behavior patterns to develop WMCID that are generic to any LMS. Precision and fairness are measured through the performance metrics described later in the next chapter. The remainder of this section will discuss the directions investigated to design and develop the WMCID approaches. This is followed by a sub-section on each broad approach used by this research: classification, optimization and hybrids.

The process used to develop the WMCID approaches was essentially identical to that used to develop the LSID approaches. A literature review was done to find an existing automatic approach to improve using CI algorithms. Although DeWMC (Chang et al., 2013) was the only approach found, it did suit the needs of this research well as it uses generic behavior patterns and was reasonably effective (80.9% accuracy). Three approaches for improving the precision of identification were considered: classification, optimization and hybrid algorithms. A classification approach was selected as such a classification algorithm may find a better function than the one used by DeWMC. DeWMC, like DeLeS, uses an unweighted average of hint values, or a set of weights each with value 1.0, to calculate the WMC for a student in a given learning session. An optimization approach is reasoned to be able to find a more optimal set of weights and so improve WMC identification precision. Although the first two approaches were expected to improve results, hybrid algorithms can overcome the weaknesses of mono-CI algorithms and so improve results further. Since there are many possible hybrids to choose from, first the classification and optimization approaches were evaluated and then a hybrid algorithm was selected based on analyzing the results.

3.4.1 Improving WMC Identification through Classification

For this research, the ANN is reasoned to be a rational choice of classification algorithms for two reasons. As with the LSID approach development, the size of the data set suggests that statistical classifiers may not work as well (Niles et al., 1989) and ANNs are capable of representing very complex functions (Hornik et al., 1990; Mitchell, 1997a).

The behavior pattern data for WMC is separated into activations and non-activation. In DeWMC (Chang et al., 2013), both activated and non-activated behavior patterns are used as an indicator of WMC. In essence, the activated behavior patterns are positive evidence for either high or low WMC depending on the particular behavior pattern (as described in section 3.2), while non-activated behavior patterns are negative evidence. As seen in Dorca et al. (2013), for WMCID-ANN there exist three strategies.

1. Consider only the positive evidence,
2. Consider only the negative evidence; or,
3. Consider both the positive and negative evidence.

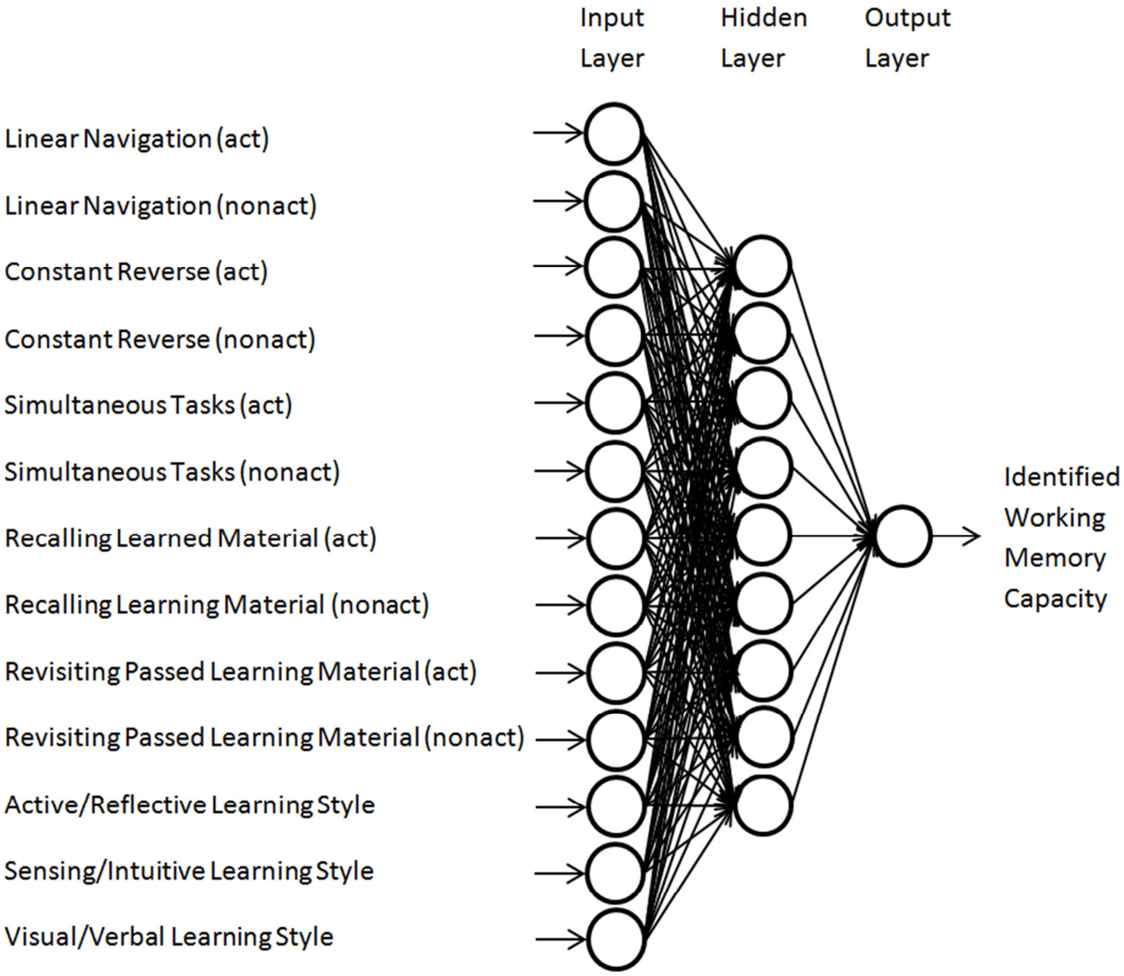
The chosen strategy determines what data is used as inputs to the ANN. For strategy 1, only the activated behavior pattern data is used, for strategy 2 only the non-activated behavior pattern data is used and for strategy 3 both are used. All three of these strategies were fully evaluated (i.e. all parameters optimized, all overfitting reduction strategies investigated and a final result produced).

With the inputs determined, the next step was to investigate if there is any need for pre-processing/normalization. Unlike for with the learning styles behavior data, the WMC behavior pattern data has both fewer extreme outliers and most of the behavior data has similar scale. The minimum value for all behavior patterns is zero and the maximum values ranges from 40 for the non-activations of *simultaneous tasks* to 472 for activations of *recalling information*. By comparison, the range of maximum values in the learning styles data ranges from 15 for the *forum_posts* behavior to 29441 for the *exercise_stay* behavior. In DeWMC (Chang et al., 2013), the three learning style indicators are combined into one by averaging them; however, for WMCID-ANN to provide more granularity the A/R, S/I and V/V learning style values were used as separate inputs. Since the learning style values are real values from 0 to 1, they were multiplied by 100 to bring them more in line with the other behavior data values. A full evaluation was performed using both normalization and no pre-processing.

The output for WMCID-ANN is straightforward. As with DeWMC (Chang et al., 2013), a real value from 0 to 1 is produced and this is decoded on a linear scale from 0 to 60 as the OpTotal value from WebOSPAN (Lin, 2007) which as previously discussed is taken as the WMC value. The topology using strategy 3 for WMCID-ANN is shown in Figure 22.

As with LSID-ANN, WMCID-ANN uses back propagation as the training method as the training data contains the actual WMC value. The error (*e*) is computed as the difference between the actual WMC and the identified WMC. The fitness of WMCID-ANN is calculated as the average of the precision (1-*e*) over all samples. The termination condition for WMCID-ANN is identical to that for LSID-ANN.

Figure 22. WMCID-ANN topology using all possible inputs


Input Layer
Hidden Layer
Output Layer
Linear Navigation (act)
Linear Navigation (nonact)
Constant Reverse (act)
Constant Reverse (nonact)
Simultaneous Tasks (act)
Simultaneous Tasks (nonact)
Recalling Learned Material (act)
Recalling Learning Material (nonact)
Revisiting Passed Learning Material (act)
Revisiting Passed Learning Material (nonact)
Active/Reflective Learning Style
Sensing/Intuitive Learning Style
Visual/Verbal Learning Style
Identified Working Memory Capacity

3.4.2 Improving WMC Identification through Optimization

This sub-section describes how optimization algorithms were adapted to improve the precision of WMC identification. As for learning styles, three optimization algorithms, ant colony system, genetic algorithm and particle swarm optimization were selected to build three corresponding approaches WMCID-ACS, WMCID-GA and WMCID-PSO. As previously described, DeWMC (Chang et al., 2013) works by calculating from student behavior pattern data a set of hint values (*h*) that are in turn averaged into a WMC value. As there is no information available on the relative importance of any behavior pattern towards calculating WMC, DeWMC gives each hint a weight of 1.0 when averaging. Although this is a reasonable assumption given the lack of information, it is unlikely that these weights are optimal. The WMCID approaches aim to enhance precision by finding a set of optimal weights for the behavior patterns. As with the LSID approaches, the solution space for this problem is fairly large at 10^{12} combinations; therefore, it was decided to investigate using optimization algorithms as they search large spaces more efficiently than brute force algorithms (Russell & Norvig, 2010). Based on literature, ACS, GA and PSO have all been successful at finding optimal weights (Abido, 2002; Ericsson et al., 2002; Pothiya et al., 2010) and so were selected as the searching algorithms. In the next paragraph, the method used by DeWMC to identify WMC is described in greater detail.

DeWMC (Chang et al., 2013) identifies WMC in the following manner. For each learning session (the time between when a student logs into the LMS and then logs out), the total number of activations (*act*) and non-activations (*nonact*) for each behavior pattern is tracked and recorded. Additionally, if a pattern is activated or non-activated during the session it is considered detected and the number of patterns detected is also recorded (*d*). The WMC hint (*h*) for a behavior pattern is the division of activations by the sum of activations and non-activations (Formula 13). An additional hint (h_{LS}) is calculated, as described Section 3.2.6, based on the student's learning styles, specifically their A/R, S/I and V/V preferences. The learning styles hint value is applied to each learning session so that it will have as much importance as the hints calculated from the navigational patterns. Since a student's learning styles do not change very much the learning styles hint value is taken as constant across all learning session.

Assuming *n* sessions, the WMC for the i^th^ learning session (*WMC$_i$*) is calculated based only on learning styles and the behavior patterns detected during that learning session. Assuming a number of detected patterns (*d*), the hint for the d^th^ detected behavior h_d and the weight for the hint is W_d (which for DeWMC is always 1.0). Formula 14 shows how WMC$_i$ is calculated from the sum of the hints including the learning styles hint (h_{LS} weighted by W_{LS} and divided by the number of hints (d+1 for learning styles).

Once a WMC value has been calculated for each session on a student-by-student basis, a weight value for the session (*Si*) is calculated as the total number of activations for that session divided by the total number of activations across all sessions as shown in Formula 15. Lastly, the WMC for each student (*WMC$_{id}$*) is determined by a weighted average of the WMC values across all sessions as shown in Formula 16. In a real world setting, these last two steps, calculating the session weights and identifying the WMC, would be completed after each session; however, in this research, the data was gathered and all calculations made after the end of the course.

$$h = \frac{act}{act + \ nonact} \tag{13}$$

$$WMC_i = \frac{W_{LS} \times h_{LS} + \sum_{x=1}^{d} W_x \times h_x}{d+1} \tag{14}$$

$$S_i = \frac{act_i}{\sum_{x=1}^{n} act_x} \tag{15}$$

$$WMC_{id} = \frac{\sum_{x=1}^{n} S_x \times WMC_x}{n} \tag{16}$$

As mentioned above, the weight values for each hint are assumed to be 1.0. This research investigated finding optimal weight values for each of the behavior and thereby improving the precision of the identified WMC value for each session (Formula 14) which in turn improves precision of the overall calculation (Formula 16).

The process for adapting each individual algorithm towards finding these weights is identical to the LSID approaches. Unlike for WMCID-ANN where the behavior pattern data could be separated into 13 inputs, for the approaches using optimization algorithms six weights are found as these approaches are optimizing the averaging formula (Formula 16) used by DeWMC (Chang et al., 2013) which has six hint values (five from the navigation patterns and one from learning styles). When using ACS a graph is created for the ants to traverse with six layers with each layer having 100 nodes assigned values iteratively from 0.01 to 1.00 in increments of 0.01. For WMCID-GA, the genome consists of six genes and each gene is limited to values from 0.01 to 1.00. With the PSO, a hypercube with six dimensions each bounded from 0.01 to 1.00 is used.

3.4.3 Improving WMC Identification through Hybrid Algorithms

This sub-section discusses the two approaches used to identify WMC using hybrid algorithms. EANN and EANN/R are discussed as the first approach as they are very similar. The second approach presented is the SISO architecture as used for learning styles.

On examination of the results from the mono-CI algorithm-based approaches, it was clear that the ANN produced the best results. Thus, it was reasoned that the focus for choosing a hybrid should be on overcoming a weakness of the ANN. As previously discussed, one of the weaknesses of the ANN is the topology forces a particular relationship style between the inputs and outputs. The EANN aims to search for an optimal topology by using an evolutionary algorithm, with genetic algorithm being a typical choice (Belew et al., 1990; Yao, 1999) and in so doing produce a more precise function describing the relationship between behaviors (with learning styles) and WMC. Since it is difficult to know if a recursive or non-recursive topology is optimal, both a recursive and non-recursive EANN were evaluated by building and evaluating two approaches called, WMCID-EANN and WMCID-EANN/R respectively.

As with other approaches that use an ANN, the first step is to decide what data will be used as inputs to the ANN. As will be shown in the results section, using both the positive and negative evidence (strategy 3) as inputs produced the best results for WMCID-ANN and so these the same 13 inputs are used for both the EANN and EANN/R. The second step is to consider what, if any, pre-preprocessing/normalization needs to be done to the input data. WMCID-ANN performed best (again as shown in the results) when the data was not normalized since the data, except the learning styles inputs, is of somewhat similar scale (a low range of 0 to 40 and a high range of 0 to 472) with almost no outliers. As the learning styles range from 0 to 1, to bring them into a similar range they were multiplied by 100. The output from WMCID-EANN and WMCID-EANN/R is the same as for WMCID-ANN, a real value from 0 to 1 which is mapped linearly to 0 to 60 is decoded as an OpTotal which determines the WMC of the student (Lin, 2007).

Before designing the genome structure to represent the topology, a training method must be picked between evolutionary and hybrid training. The hybrid training model has the advantage of using the GA as a global search mechanism and backpropagation as a local search, with the main drawback of increased training time compared to evolutionary training. Hybrid training was selected as the average training time was less than 30 minutes on an Intel i7-4770 which is not unreasonable.

For a non-recursive EANN using hybrid training needs a genome which can describe all possible unweighted non-recurrent topologies. For WMCID-EANN, the same format is used as described by Yao (1999) for representing such a genome described in section 2.6.6. For an ANN of I+H+O nodes, where I is the number of inputs (13), H is number of hidden nodes (the number of hidden nodes is a control parameter) and O is the number of output nodes (1), the first input node requires I+H+O-1 genes (-1 since it cannot connect to itself). The second input node requires I+H+O-2 nodes (-2 since it cannot connect to itself or the 1st node). Since the hidden nodes cannot connect to an input node in a non-recursive EANN, the 1st hidden node requires H+O-1 nodes and then a similar pattern is followed. A similar pattern would be followed for the output nodes except in this case there is only 1 so it cannot connect to anything in a non-recursive EANN and no genes are required. Thus, the formula for the number of genes needed for a non-recursive EANN is shown in Formula 17. The first term represents the connection between inputs, where the 1st input may be connected to the 2nd, 3rd, 4th, etc. and the 2nd input may be connected to the 3rd, 4th, etc. The second term represents the connection between the hidden nodes. The third and fourth terms represent the possible connections from the input nodes to the hidden and output nodes respectively. The fifth and final term describes the connections from the hidden nodes to the output nodes. For EANN/R, the genome needs to describe all of the possible connections between any two nodes including itself which is simply the number of nodes (I+H+O) squared (shown in Formula 18).

$$G_{EANN} = \frac{I(I-1)}{2} + \frac{H(H-1)}{2} + I \times H + I \times O + H \times O \tag{17}$$

$$G_{EANN/R} = (I + H + O)^2 \tag{18}$$

For this research, it was decided that no node should be completely cut out from the ANN (i.e. have no inputs or no outputs). For the input node this is justified by the assumption that all of the behavior data is relevant and should therefore not be excluded. The decision was made to precisely control the number of hidden nodes in the network by a control parameter rather than let it vary. For the output node, it clearly cannot be excluded. To find an invalid topology, the rules R1 and R2 were included into the algorithm and all invalid genomes are given the minimum fitness value of 0.

> R1: All hidden and output nodes must have at least one input from a different node
>
> R2: All hidden and input nodes must have at least one output to a different node

For genomes which produce a valid topology, the fitness of the genome is equal to the fitness of the resulting ANN. The fitness of the ANN is calculated exactly as WMCID-ANN. In other aspects the GA portion of the EANN uses the same operators and processing as for LSID-GA and WMCID-GA summarized as follows. The population is initialized to size P. From the population P/2 unique pairs are selected using the roulette wheel technique. The uniform crossover operator is applied to each pair to produce new offspring. The uniform mutation operator is applied to each new offspring. If no change has occurred as a result of the two operators, a single gene is forced to mutate. The new offspring are merged into the population and an elitist survival strategy removes those with the lowest fitness until the population is of size P again.

With respect to using a loosely couple hybrid architecture (i.e. SISO), the analysis of the results for WMC did not indicate that there were multiple good solutions as there were for learning styles. If any WMCID approach was re-trained, then the results were relatively consistent for each individual student; whereas, for learning styles re-training would produce very different individual results (as described in Section 3.3.3). Thus it is reasoned that if splitting the data is effective because there are multiple good solutions for identifying learning styles then such an architecture should not provide much improvement when there is no indication of multiple good solutions. Thus, an approach called WMCID-SISO was built and evaluated using SISO-style architecture to confirm that the reasoning to use it for learning styles identification was justified. Just as with LSID-SISO, two different algorithms were evaluated for the *Prediction* step, GA, as the best mono-optimization algorithm, and an ANN. As with LSID-SISO, the *Confidence* and *Solve* steps use ANNs as the ANN was found to be best overall. These two versions were called WMCID-SISO (GA) and WMCID-SISO (ANN) (shown in Figures 23 and 24). In all other aspects, WMCID-SISO was developed and evaluated exactly as LSID-SISO.

Figure 23. WMCID-SISO (GA) Architecture

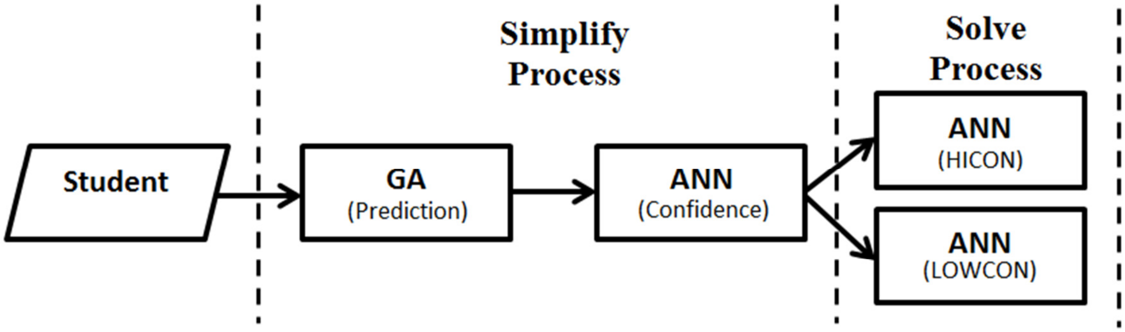
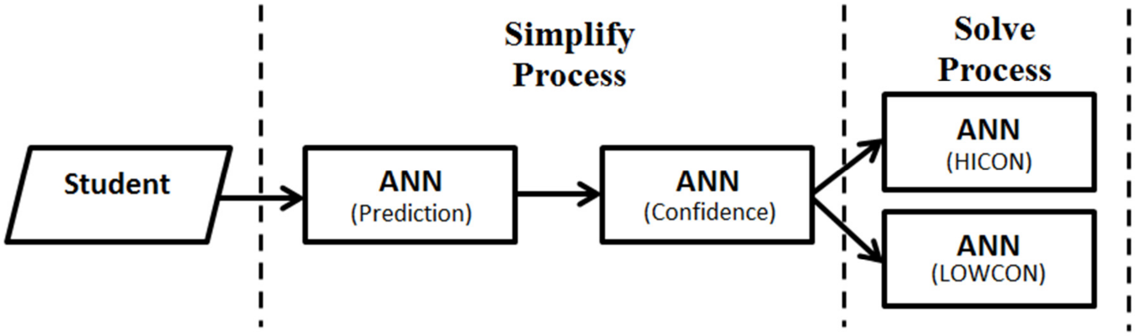


Figure 24. WMCID-SISO (ANN) Architecture



Chapter IV - Methods

This chapter explains the methodology to evaluate the LSID and WMCID approaches. First the data sets are described and compared in size (where possible) with those of related works. This is followed by a discussion on the 10 fold cross validation process used to ensure that the algorithms are generalized. The performance metrics used to evaluate the LSID and WMCID approaches are then explained. A discussion is presented on parameter optimization process used for the CI algorithms. The chapter concludes with a look at overfitting reduction strategies used for the LSID and WMCID.

4.1 Training Data

The training data for this research consists of two data sets, one set for learning style identification and another for WMC identification. First, the data used for the LSID approaches will be discussed followed by the WMCID data set. For each data set, the overall data set is described and then any removals from the data set are discussed. For the LSID data set, it is compared in size to related works. As there are no related works for WMCID, no comparison is made.

4.1.1 LSID Training Data

The LSID data set is the same data set used by Graf et al. (2009) for evaluating DeLeS. The data set consists of both behavior data and the student’s actual learning styles (as identified by the ILS (Felder & Solomon, 1998)) for 127 students from an undergraduate computer science / information technology course. In order to ensure the identified learning styles are reliable, any student who spent less than 5 minutes filling in the questionnaire was eliminated from the data set. In addition, to ensure there is sufficient data about each student, only students who submitted more than half of the assignments and took the final exam were used. After these removals, the final dataset consists of 75 students. This data set is of similar size to that found in the other existing approaches that are described as follows. Latham et al. (2012) conducted six experiments with data set sizes of 75, 75, 89, 76, 94 and 95. García et al. (2007) had 77 students in their data set (50 for training and 27 for testing). The data set is larger than that used by Özpolat and Akar (2009) with 40 students (10 for training and 30 for testing) and Cha (2006) with between 23 and 49 students as theirs varied by FSLSM dimension with correspondingly different sizes of training and testing data sets.

To ensure that the LSID data set fairly represents learning styles for students the distribution of learning styles is examined. Table 7 shows the percentage of students in the data set with an active, sensing, visual or sequential learning style (shown in the "LSID" row) and the range of values found in literature (Felder & Spurlin, 2005). It can be seen that the distribution of learning styles for the data used by this research is well within the range of expected values.

Table 7. Comparison of the distribution of learning styles

	Active	Sensing	Visual	Sequential
LSID	52.7%	63.0%	81.9%	53.4%
Felder & Spurlin (2005)	47-70%	46-81%	66-91%	45-85%

4.1.2 WMCID Training Data

The WMCID approaches are evaluated using a data set consisting of behavior pattern data and learning styles data (as identified by the ILS (Felder & Solomon, 1998)) and each student's actual WMC (as identified using WebOSPAN) from 75 undergraduate students. As with the LSID data set, in order to ensure that only high quality data is used students who spent less than five minutes on the ILS (Felder & Solomon, 1998) or students who had more than 15 errors in WebOSPAN were removed. This resulted in a final data set of 63 students. As no other automatic WMC identification algorithm could be found in literature, no comparison of the size of the data set is possible.

4.2 Ten Fold Cross Validation

A 10 fold cross validation process is used for control parameter optimization, evaluating overfitting reduction and producing a final result to ensure that both LSID and WMCID are generalized by exposing the approaches to different data sets. With the 10 fold cross validation process, the algorithm is executed 10 times with the results averaged over the 10 executions each with a different training and assessment data sets, i.e. a fold, extracted from the overall data set. For each fold, 1/10th of the students are selected for the assessment set and chosen such that each student is selected for only a single fold's assessment set; thereby, guaranteeing that each assessment set is unique. The remaining unselected students are used as the training data for the fold.

4.3 Performance Metrics

The performance metrics used for LSID and WMCID are very similar, with LSID having an extra metric. The metrics are used in every step of the evaluation, i.e. for control parameter optimization, evaluating overfitting strategies and producing a final result. The shared metrics are accuracy (ACC), lowest accuracy (LACC) and percentage of students identified with reasonable accuracy (%Match). LSID also uses the similarity metric (SIM) as it is used commonly in literature for learning styles identification (García et al., 2007; Graf, Kinshuk et al., 2009; Özpolat & Akar, 2009); however, no corresponding metric could be found for WMC. ACC and SIM measure average performance, while LACC and %Match measure performance on a student-by-student basis. The remainder of this section will discuss SIM first, followed by ACC, LACC and %Match.

SIM works by dividing learning styles values, ranging from 0 to 1, into three regions: an upper region (>0.75) (LS_U), a lower region (<0.25) (LS_L) and a balanced region (>=0.25 and <= 0.75) (LS_B). If a student's actual learning styles value (LS_{actual}) is in the same region as the identified value (LS_{id}) then SIM returns 1; when the two values are in adjacent regions SIM returns 0.5 and when they are in opposite regions SIM returns 0 (shown in Formula 19) where R is a function returning the region of a learning style value. SIM values are calculated for each student and then averaged to measure the precision of an algorithm over the whole student population.

$$SIM = \begin{cases} 1.0 & if\ R(LS_{id}) = R(LS_{actual}) \\ 0.5 & if R(LS_{id}) \neq R(LS_{actual})\ and\ (R(LS_{id}) = LS_B\ or\ R(LS_{actual}) = LS_B) \\ 0.0 & otherwise \end{cases} \tag{19}$$

Although SIM is commonly used and is suitable for algorithms such as Bayesian algorithm which return a classification (e.g., returning active, balanced or reflective for the A/R dimension), it does have a notable drawback of reduced accuracy for approaches which can return concrete values (such as LSID, WMCID and DeLeS). The drawback occurs largely when the LS_{actual} and / or the $LS_{identified}$ are close to the region edges. For example, if LS_{actual} = 0.76 and LS_{id} = 0.74 then SIM returns a 0.5 (moderate match) even though this is a very close match. Although, there is no SIM-like metric for WMC, and so no corresponding issue, using the ACC is still reasonable as WMCID returns a concrete value. So to measure the performance of LSID and WMCID, the exact difference between actual and identified learning styles or WMC values is used as shown in

Formula 20 and 21 respectively. The ACC values are calculated for each student and then averaged to measure the precision for the algorithm.

$$ACC_{LSID} = 1 - |\, LS_{actual} - \, LS_{id} \,| \tag{20}$$

$$ACC_{WMCID} = 1 - |\, WMC_{actual} - \, WMC_{id} \,| \tag{21}$$

Since students can be negatively affected by mismatched content or inappropriate interventions from a teacher (Graf, Chung et al., 2009; Kirschner, 2002; Paas et al., 2004; Van Merriënboer et al., 2002) it is desirable that any amount of misidentification be minimized. So in addition to the average metrics describe above, the results are examined on an individual student basis. To measure the performance of an approach with respect to individual students two metrics are used. For each of these metrics' formulae the following definitions and assumptions are made, a data set of size n students is assumed where $LS_{actual,x}$ and $LS_{id,x}$ are the actual and identified learning styles for the x^{th} student in the data set. Similarly, $WMC_{actual,x}$ and $WMC_{id,x}$ are the actual and identified WMC values for the x^{th} student in the data set. The first metric is the lowest ACC (LACC) value calculated for any student as shown in Formulae 22 and 23. LACC measures the worst case scenario for any student. The second metric (%Match) is the percentage of students matched with reasonable precision which is defined as within half of the range of possible values (shown in Formula 24 and 25). For both learning styles and WMC, a student is considered matched if the ACC is within half of the range of possible values in the data set for that characteristic. For learning styles, the range of values extends from 0 to 1, so for learning styles, a student is matched if the ACC ≥ 0.5. For WMC, the range of values was 0.233 to 0.900. Half of this range is 0.333 which was rounded to 0.3 giving a match condition for WMC of ACC ≥ 0.7.

$$LACC_{LSID} = \min_{1 \le x \le n} ACC(LS_{actual,x}, LS_{id,x}) \tag{22}$$

$$LACC_{WMCID} = \min_{1 \le x \le n} ACC(WMC_{actual,x}, WMC_{id,x}) \tag{23}$$

$$\%Match_{LSID} = \frac{\sum_{x=1}^{n} \begin{cases} 0.0 & if ACC\left(LS_{actual,x}, LS_{id,x}\right) < 0.5 \\ 1.0 & otherwise \end{cases}}{n} \tag{24}$$

$$\%Match_{WMCID} = \frac{\sum_{x=1}^{n} \begin{cases} 0.0 & if ACC\left(WMC_{actual,x}, WMC_{id,x}\right) < 0.7 \\ 1.0 & otherwise \end{cases}}{n} \tag{25}$$

4.4 Parameter Optimization

The process to optimize the parameters for the CI algorithms is described as follows and differs only by the parameters and default values for each algorithm. As a first step, a literature review was performed to find either a suitable range or principles for each parameter, resulting in a set of values for each parameter. A mid-range value was selected from the set of values to act as a default value (shown in bold) with the exception of the VMax parameter for PSO as the recommended value is the highest possible value (Shi & Eberhart, 1998). The algorithm is executed iteratively cycling through every value in the set for the first parameter with the remaining parameters using their default value. The parameter value which produces the best result is considered the optimal setting and used for all subsequent executions. The process is then repeated for each parameter.

4.4.1. Parameter Optimization for Artificial Neural Networks

The control parameters for the ANN are optimized in the following order: number of hidden node (*H*), learning rate (*η*), momentum (*m*) and training mode. Literature suggests an *H* value between log T (where T is the size of the training set) (Wanas, Auda, S. Kamel, & Karray, 1998) and 2 × the number of inputs (Swingler, 1996). In this case, the lower bound is log 67 or 1.82 for the LSID approaches and log 57 or 1.75 for the WMCID approaches. To maximize the changes of optimization the lower bound is reduced to 1 instead of rounded up to 2. The upper bound varies for each learning style dimension and WMC since the number of behavior patterns (inputs) varies (24 for A/R, 26 for S/I, 12 for V/V, 18 for S/G and 26 for WMC). For learning rate, a low value is suggested (Swingler, 1996) so the values evaluated were (0.001, 0.01, 0.01, 0.02, 0.03, 0.04, **0.05**, 0.06, 0.07, 0.08, 0.09, 0.1). Momentum is also recommended to be low so that it does not cause the ANN to skip past good areas during training (Swingler, 1996). So the values evaluated for momentum are (0.00 0.01, 0.02, 0.03, 0.04, **0.05**, 0.06, 0.07, 0.08, 0.09, 0.1). Both individual and ensemble training modes are evaluated and with individual used as the default. The optimal parameters for algorithms which use an ANN are shown in Tables 8 to 19. The optimal parameters for the prediction step of LSID-SISO (ANN) and WMCID-SISO (ANN) are the same as LSID-ANN and WMCID-ANN respectively as they are the same algorithm.

Table 8. Optimal parameter settings for LSID-ANN, LSID-SISO (ANN), Prediction Step

	H	η	m	Training Mode
A/R	1	0.08	0.10	Individual
S/I	5	0.06	0.09	Individual
V/V	8	0.08	0.06	Individual
S/G	2	0.07	0.01	Individual

Table 9. Optimal Parameter Values for LSID-SISO (ACS), Confidence Step

	H	η	m	Training Mode
A/R	2	0.08	0.01	Individual
S/I	8	0.03	0.01	Individual
V/V	3	0.02	0.05	Individual
S/G	9	0.01	0.02	Individual

Table 10. Optimal Parameter Values for LSID-SISO (ANN), Confidence Step

	H	η	m	Training Mode
A/R	3	0.07	0.01	Individual
S/I	8	0.05	0.04	Individual
V/V	6	0.06	0.02	Individual
S/G	7	0.03	0.00	Individual

Table 11. Optimal Parameter Values for LSID-SISO (ACS), Solve Step

		H	η	m	Training Mode
A/R	HICON	2	0.08	0.01	Individual
	LOWCON	5	0.06	0.02	Individual
S/I	HICON	2	0.04	0.03	Individual
	LOWCON	7	0.03	0.02	Individual
V/V	HICON	3	0.02	0.05	Individual
	LOWCON	3	0.03	0.03	Individual
S/G	HICON	8	0.01	0.00	Individual
	LOWCON	7	0.04	0.03	Individual

Table 12. Optimal Parameter Values for LSID-SISO (ANN), Solve Step

		H	η	m	Training Mode
A/R	HICON	4	0.06	0.01	Individual
	LOWCON	3	0.05	0.01	Individual
S/I	HICON	5	0.03	0.04	Individual
	LOWCON	6	0.03	0.01	Individual
V/V	HICON	3	0.03	0.03	Individual
	LOWCON	3	0.06	0.02	Individual
S/G	HICON	7	0.02	0.01	Individual
	LOWCON	9	0.04	0.01	Individual

Table 13. Optimal Parameter Values for WMCID-ANN and WMCID-SISO (ANN), Prediction Step

H	η	m	Training Mode
3	0.001	0.07	Individual

Table 14. Optimal ANN Parameter Values for WMCID-EANN

H	η	α	Training Mode
3	0.001	0.05	Individual

Table 15. Optimal ANN Parameter Values for WMCID-EANN/R

H	η	m	Training Mode
8	0.01	0.03	Individual

Table 16. Optimal Parameter Values for WMCID-SISO (GA), Confidence Step

H	η	m	Training Mode
3	0.001	0.07	Individual

Table 17. Optimal Parameter Values for WMCID-SISO (ANN), Confidence Step

H	η	m	Training Mode
4	0.001	0.04	Individual

Table 18. Optimal Parameter Values for WMCID-SISO (GA), Solve Step

	H	η	m	Training Mode
HICON	3	0.001	0.07	Individual
LOWCON	5	0.01	0.04	Individual

Table 19. Optimal Parameter Values for WMCID-SISO (ANN), Solve Step

	H	η	m	Training Mode
HICON	5	0.02	0.03	Individual
LOWCON	5	0.05	0.00	Individual

4.4.2. Parameter Optimization for Ant Colony System

The following parameters are optimized for the ACS: population size (*P*), local quality weight (α), pheromone weight (β), evaporation ratio (ρ), consumption ratio (τ_0), exploitation factor (q_0). The recommended population size varies from 10 to 100 (Aghdam, Ghasem-Aghaee, & Basiri, 2009; Dorigo & Gambardella, 1997a; Huang, 2001; Maier et al., 2003; Shmygelska & Hoos, 2005); however, this is expanded to maximize the chance of optimization giving the set of values (10, 25, 50, 100, 200).

Table 20. Optimal Parameter Values for LSID-ACS and LSID-SISO (ACS), Prediction Step

	P	ρ	τ_0	q_0
A/R	100	0.80	0.20	0.00
S/I	200	0.80	0.20	0.00
V/V	50	0.90	0.05	0.00
S/G	200	0.50	0.20	0.00

Table 21. Optimal Parameter Values for WMCID-ACS

P	ρ	τ_0	q_0
100	0.80	0.03	0.80

As discussed previously, since there is no information available to determine the local quality for any particular pattern weight, they are set to 1.0 for each link. This means that pheromone will quickly dominate every ant's decision so there is no need to optimize the α and β parameters, so both are set to 1.0. The evaporation ratio influences the amount of exploration vs. exploitation and so is problem specific but is generally preferred to be somewhat high (Dorigo & Gambardella, 1997b). The evaporation ratio values evaluated are (0.5, 0.6, **0.7**, 0.8, 0.9). Like the evaporation ratio, the consumption ratio parameter influences the preference of exploration vs. exploitation, and it is generally preferred to be lower as if every ant consumes a lot of pheromone then there will quickly be none left at all. The consumption ratio values evaluated are (0.01, 0.05, **0.10**, 0.20, 0.30) (Dorigo & Gambardella, 1997b). The exploitation factor is generally preferred to be high so that the ants will use previously found good solutions (Dorigo & Gambardella, 1997b); however, the lack of a local quality is a significant change to how the pseudorandom proportional rule is intended to work, so the exploitation parameter is also evaluated as being off ($q_0 = 0.0$) giving the set of values (0.0, 0.5, 0.6, **0.7**, 0.8, 0.9).

The optimal parameter values computed for algorithms which used ACS are shown in Tables 20 to 21. The prediction step parameters for LSID-SISO (ACS) are identical to LSID-ACS as they are the same algorithm.

4.4.3. Parameter Optimization for Genetic Algorithm

The control parameters optimized for the GA are: population size (*P*), crossover weight (*C*) and mutation weight (*M*) and the parameter optimization principles that follow were found from reviewing literature (Grefenstette, 1986; Srinivas & Patnaik, 1994). In general, for larger populations it is suggested to use lower crossover and mutation weights (Grefenstette, 1986) and vice versa; however, since no firm relationship is known between these parameters a variety of configurations is evaluated. Grefenstette (1986) recommends either very low or very high populations. His work examined up to 160 genomes (but does not recommend this as a strict upper limit) so we expand on this upper limit to 200 in order to maximize the chances of finding an optimal setting. This gives a set of population values of (25, 50, 100, 150, 200). Crossover weight is recommended to be above 0.6 (Srinivas & Patnaik, 1994) and so the set of values evaluated is (0.6, 0.7, **0.8**, 0.9). Finally mutation weight is generally preferred to be less than 0.05 (Grefenstette, 1986; Srinivas & Patnaik, 1994) giving a set of values of (0.0001, 0.001, 0.01, 0.02, **0.03**, 0.04, 0.05). The optimal parameters for all algorithms which use a GA are shown in Table 22 to 25. The parameters for the prediction step of WMCID-SISO (GA) are the same as WMCID-GA as they are the same algorithm.

Table 22. Optimal Parameter Values for LSID-GA

	P	C	M
A / R	200	0.80	0.03
S / I	100	0.90	0.03
V / V	100	0.80	0.03
S / G	400	0.70	0.04

Table 23. Optimal Parameter Values for WMCID-GA and WMCID-SISO (GA), Prediction Step

P	C	M
25	0.80	0.001

Table 24. Optimal Parameter Values for WMCID-EANN

P	C	M
25	0.70	0.01

Table 25. Optimal Parameter Values for WMCID-EANN/R

P	C	M
25	0.70	0.01

4.4.4. Parameter Optimization for Particle Swarm Optimization

The following PSO parameters are optimized to ensure proper functioning: population size (*P*), acceleration coefficients (*c1* and *c2*), inertia (*w*) and maximum velocity (*VMax*). As with the other algorithms, a literature review (Clerc & Kennedy, 2002; Eberhart & Kennedy, 1995; Shi & Eberhart, 1998) was conducted to find recommended ranges for each parameter. As with the population size parameter for the ACS and GA, it was expanded from the recommended value of 100 or less (Clerc & Kennedy, 2002; Eberhart & Kennedy, 1995; Shi & Eberhart, 1998) to 400 thereby giving the set: (25, 50, 75, 100, 200, 400). The acceleration coefficient for the individual best (*c1*) was given the values (0.0, 0.25, **0.5**, 0.75, 1.0). Since the global best position must always be considered, the acceleration coefficient (*c2*) was given the values (0.25, **0.5**, 0.75, 1.0). The recommended range for inertia is 0.9 to 1.2 (Shi & Eberhart, 1998); however, this was expanded slightly to allow for a greater chance of optimization giving this set of values: (0.75, 0.9, **1.0**, 1.1, 1.2). For bounded problems, a good initial value for the maximum velocity (*Vmax*) is recommended to be set equal to the extent of the bounds (*Xmax*) as it is likely optimal; however, a trial-and-error process is recommended to refine the value (Shi & Eberhart, 1998). The bounds for the problem in this research are the minimum (0.01) and maximum (1.0) weights, giving an *Xmax* = 0.99. Therefore, the default value for *Vmax* is set to 0.99 instead of a mid-range value. There is no reason to assess a value of *Vmax*>*Xmax* as this has the same effect as *Vmax*=*Xmax.* If a particle has a velocity (*v*) such that *v*≥*Xmax,* it simply hits the hypershape boundary. However, there is a reason to assess *Vmax*≤*Xmax* as this promotes exploitation of promising areas by preventing particles from flying away from them too quickly. A set of coefficients (0.05, 0.1, 0.25, 0.5 and 1.0) were selected focusing on lower values to evaluate keeping the particles closer to promising areas. The set of *Vmax* values was found by multiplying *Xmax* by the each of the coefficients giving a set of value for *Vmax* of (0.0495, 0.099, 0.2475, 0.495, **0.990**). The optimal parameters for approaches using PSO are shown in Table 26 and 27.

Table 26. Optimal Parameter Values for LSID-PSO

	P	c1	c2	w	Vmax
A / R	400	1.00	1.00	0.75	0.990
S / I	100	0.25	1.00	1.20	0.990
V / V	400	0.50	1.00	1.00	0.099
S / G	50	1.00	1.00	0.90	0.495

Table 27. Optimal Parameter Values for WMCID-PSO

P	c1	c2	w	Vmax
100	1.0	1.0	1.1	1.0

4.5 Overfitting Reduction Strategies

With CI algorithms, overfitting is a common problem where the solution is fit to noise in the data and is not generalized to future data. Such overfitting reduces the quality of the found solution, i.e. reduces the precision of learning styles/WMC identification. Fortunately, techniques exist which can reduce overfitting and three such techniques were investigated for this study: stratification (Kohavi, 1995), future error prediction (FEP) (Mitchell, 1997b) and weight decay (Krogh & Hertz, 1992).

Stratification works by ensuring the training sets and assessment sets are picked such that they have a similar distribution to expected future data sets (Kohavi, 1995). Thus, even if the algorithm's solution is overfit it will be overfit to likely future data and so will be effective anyway. As seen in section 4.1.1., the data used in this research has a distribution well within the range of values found by Felder and Spurlin (2005). Therefore, it can be assumed that future data will be distributed similarly. To ensure that each training set and assessment set has a proper distribution students are first grouped in accordance to their preference (e.g., all students with an active preference are grouped together). Then for each fold's assessment set, students are picked so that the percentage of students in the assessment set with a particular preference is as close as possible to the actual percentage with that preference. For example, for an assessment set consisting of 7 students for the A/R dimension, 4 active students are selected as this gives the closest possible percentage to 52.7%. The remainder are selected from the other preference, i.e. 3 students with a reflective preference. With respect to WMC, a common means to separate high and low WMC is using a split along the median (Beilock & Carr, 2005; Schmeichel, Volokhov, & Demaree, 2008; Tuholski, Engle, & Baylis, 2001). Thus, our data set was split along the median into high and low WMC sets and an equal number of high and low WMC students were selected for each assessment set. With 63 students this meant that there were seven assessments set of size 6, and three of size 7. For the size 6, sets three students with high and three with low WMC were selected. For the two of the size 7 sets, four students with high WMC were selected and three with low WMC. For the last assessment set, four students with low WMC were selected and three with high WMC. For both learning styles and WMC, all students not in the assessment set are put into the training set. Stratification is assessed as either on or off.

FEP functions by attempting to detect when overfitting starts to occur and then terminating the algorithm (Mitchell, 1997b). This is done by extracting a validation set from the data set to represent future data samples. Whenever a new best solution is found, a result is produced using the validation set. If this result is worse than the previous result from the validation set then overfitting is assumed to be happening and the algorithm is terminated. FEP is assessed as either on or off. To prevent early termination due to chance (from unstable results) a minimum number of generations (G_{min}) must be completed. The minimum number of generations used by FEP overrides the minimum number of generation requirement used by the termination condition. As no literature could be found suggesting values for G_{min}, the values were found by examining the early relationship between fitness and precision (ACC) by seeking the point at which a higher fitness consistently produced a higher precision. It was found that stability was reached by generation 500 and often much sooner. Thus, the set of values used for G_{min} is (25, 50, 75, 100, 200, 300, 500).

The inspiration for weight decay comes from the observation that lower weight values for the neural links have been shown to be associated with better generalization in ANNs (Krogh & Hertz, 1992). With weight decay a percentage ($0 < \lambda < 1$) of the weight of each neural link is lost each generation. Therefore, a weight value on a neural link will only stay high if there is a consistent pressure from many of the samples to keep it elevated. The weight decay should be low so that the upwards pressure from training can overcome the decay when warranted. The values assessed for weight decay were (**0.00**, 0.001, 0.01, 0.02, 0.03, 0.04, 0.05).

Stratification and FEP were used for each algorithm; whereas, weight decay was only used with the ANN as it is ANN specific. Each combination of stratification and FEP were evaluated, i.e. stratification on with FEP on and each G_{min} setting and FEP off then repeated with stratification off. For LSID-ANN, weight decay was investigated first and the optimal weight decay setting was used when investigating stratification and FEP. The optimal overfitting reduction settings for each algorithm are shown in Tables 28 to 45.

Table 28. Overfitting Reduction Settings for LSID-ANN and LSID-SISO (ANN), Prediction Step

	Stratification	FEP	G_{min}	Weight Decay
A / R	On	Off	-	0.05
S / I	On	Off	-	0.05
V / V	On	Off	-	0.01
S / G	Off	Off	-	0.10

Table 29. Overfitting Reduction Settings for LSID-GA

	Stratification	FEP	G_{min}
A / R	On	On	100
S / I	On	On	25
V / V	On	On	75
S / G	On	On	25

Table 30. Overfitting Reduction Settings for LSID-ACS and LSID-SISO (ACS), Prediction Step

	Stratification	FEP	G_{min}
A / R	On	Off	-
S / I	On	Off	-
V / V	On	Off	-
S / G	On	Off	-

Table 31. Overfitting Reduction Settings for LSID-PSO

	Stratification	FEP	G_{min}
A / R	On	Off	-
S / I	On	Off	-
V / V	On	Off	-
S / G	On	Off	-

Table 32. Overfitting Reduction Settings for LSID-SISO (ACS), Confidence Step

	Stratification	FEP	G_{min}	Weight Decay
A / R	On	Off	-	0.02
S / I	On	Off	-	0.01
V / V	On	Off	-	0.01
S / G	On	Off	-	0.02

Table 33. Overfitting Reduction Settings for LSID-SISO (ACS), Solve Step

		Stratification	FEP	G_{min}	Weight Decay
A / R	HICON	On	Off	-	0.05
	LOWCON	On	Off	-	0.03
S / I	HICON	On	Off	-	0.01
	LOWCON	On	Off	-	0.00
V / V	HICON	On	Off	-	0.01
	LOWCON	On	Off	-	0.01
S / G	HICON	On	Off	-	0.01
	LOWCON	On	Off	-	0.02

Table 34. Overfitting Reduction Settings for LSID-SISO (ANN), Confidence Step

	Stratification	FEP	G_{min}	Weight Decay
A / R	On	Off	-	0.02
S / I	On	Off	-	0.02
V / V	On	Off	-	0.01
S / G	On	Off	-	0.03

Table 35. Overfitting Reduction Settings for LSID-SISO (ANN), Solve Step

		Stratification	FEP	G_{min}	Weight Decay
A / R	HICON	On	Off	-	0.03
	LOWCON	On	Off	-	0.03
S / I	HICON	On	Off	-	0.01
	LOWCON	On	Off	-	0.00
V / V	HICON	On	Off	-	0.01
	LOWCON	On	Off	-	0.01
S / G	HICON	On	Off	-	0.02
	LOWCON	On	Off	-	0.02

Table 36. Overfitting Reduction Settings for WMCID-ANN and WMCID-SISO (ANN), Prediction Step

Stratification	FEP	G_{min}	Weight Decay
On	On	30	0.00

Table 37. Overfitting Reduction Settings for WMCID-GA and WMCID-SISO (GA), Prediction Step

Stratification	FEP	G_{min}
On	On	25

Table 38. Overfitting Reduction Settings for WMCID-ACS

Stratification	FEP	G_{min}
On	On	100

Table 39. Overfitting Reduction Settings for WMCID-PSO

Stratification	FEP	G_{min}
On	On	75

Table 40. Overfitting reduction value sets and settings for WMCID-EANN

	Stratification	FEP	G_{min}	Weight Decay
GA	Off	Off	-	n/a
ANN	Off	Off	-	0.00

Table 41. Overfitting reduction value sets and settings for WMCID-EANN/R

	Stratification	FEP	G_{min}	Weight Decay
GA	Off	Off	-	n/a
ANN	On	Off	-	0.001

Table 42. Overfitting Reduction Settings for WMCID-SISO (GA), Confidence Step

Stratification	FEP	G_{min}	Weight Decay
On	Off	-	0.01

Table 43. Overfitting Reduction Settings for WMCID-SISO (GA), Solve Step

	Stratification	FEP	G_{min}	Weight Decay
HICON	On	Off	-	0.03
LOWCON	On	Off	-	0.01

Table 44. Overfitting Reduction Settings for WMCID-SISO (ANN), Confidence Step

Stratification	FEP	G_{min}	Weight Decay
On	Off	-	0.01

Table 45. Overfitting Reduction Settings for WMCID-SISO (ANN), Solve Step

	Stratification	FEP	G_{min}	Weight Decay
HICON	On	Off	-	0.04
LOWCON	On	Off	-	0.03

Chapter V - Results

This chapter will provide and compare the final results for the LSID and WMCID approaches obtained using the optimal parameter and overfitting reduction settings. The chapter is broken down into two sections with the first section presenting the results for the LSID approaches and the second section discussing the WMCID approaches.

5.1 Learning Styles Identifier

A final result with all four performance metrics is obtained for each LSID approach using the optimal parameter settings and the optimal overfitting reduction settings. First, a comparison of the SIM metric is performed between the LSID approaches, DeLeS (Graf, Kinshuk et al., 2009), a Bayesian approach (García et al., 2007) and a Naïve Bayes Tree (NBTree) approach (Özpolat & Akar, 2009) (shown in Table 46). The other related works are not included in the comparison, because they either have no evaluation (Carmona et al., 2008), use simulated data (Dorça et al., 2013) or can only classify a subset of students (Cha et al., 2006). Also, no comparison is made between LSID and Oscar (Latham et al., 2012) as their approach is tied to their LMS while LSID is general to any LMS. The comparison on SIM shows at least one LSID approach is best for each learning style dimension and all of the LSID approaches are better than the related works when averaging over all dimensions. With respect to only the related works (shown below the dotted line), DeLeS is the best (or tied in S/G) among the related works and best amongst them when averaging all dimension. So, for the remainder of the performance metrics, LSID is compared only to DeLeS as shown in Tables 47 to 49.

Table 46. Comparison of results for SIM metric

Approach	SIM				
	A/R	S/I	V/V	S/G	Average
LSID-ANN	0.802 (2)	0.741 (7)	0.727 (7)	**0.825 (1)**	0.774 (5)
LSID-ACS	**0.804 (1)**	0.762 (4)	0.771 (2)	0.785 (4)	0.781 (3)
LSID-GA	0.801 (3)	**0.781 (1)**	0.755 (5)	0.818 (2)	0.784 (2)
LSID-PSO	0.801 (3)	0.755 (6)	0.756 (4)	0.810 (3)	0.781 (3)
LSID-SISO (ACS)	0.802 (2)	0.761 (5)	**0.827 (1)**	**0.825 (1)**	**0.804 (1)**
LSID-SISO (ANN)	0.802 (2)	0.755 (6)	0.739 (6)	**0.825 (1)**	0.780 (4)
DeLeS (Graf, Kinshuk et al., 2009)	0.793 (4)	0.773 (2)	0.767 (3)	0.733 (5)	0.767 (6)
Bayesian (García et al., 2007)	0.580 (6)	0.770 (3)	-	0.630 (6)	0.660 (8)
Naïve Bayes Tree (Özpolat & Akar, 2009)	0.700 (5)	0.733 (8)	0.533 (8)	0.733 (5)	0.675 (7)

In the A/R dimension, LSID-ACS, LSID-SISO (ACS) and LSID-SISO (ANN) are tied for first in the ACC and %Match metrics. For the LACC metric, LSID-SISO (ANN) has top result; although, LSID-SISO (ACS) is not far behind. For the S/I and V/V dimensions, LSID-SISO (ACS) is the top approach for all metrics. Lastly, for the S/G dimension LSID-SISO (ACS) has the best results for ACC and %Match, and although it did well in LACC, it was behind LSID-ANN and LSID-SISO (ANN). When considering the average result across all dimensions, LSID-SISO (ACS) has the best result in each dimension.

Both LSID-SISO approaches provide an improvement over the mono-AI approaches in ACC in the S/I, V/V and S/G dimensions. This suggests that for those dimensions splitting the data set into optimal subgroups is an effective strategy. LSID-SISO (ACS) performed generally better than LSID-SISO (ANN) for most. This suggests that there is a benefit to combining ACS and ANN together, and that they capitalize on each other strengths to some degree.

Table 47. Comparison of ACC results (ranks in parentheses, top result bolded)

Approach	ACC				
	A/R	S/I	V/V	S/G	Average
LSID-ANN	0.802 (3)	0.790 (6)	0.840 (3)	0.797 (2)	0.807 (3)
LSID-ACS	**0.819 (1)**	0.797 (3)	0.799 (4)	0.737 (6)	0.788 (6)
LSID-GA	0.795 (5)	0.796 (4)	0.794 (6)	0.774 (4)	0.790 (5)
LSID-PSO	0.805 (2)	0.794 (5)	0.796 (5)	0.768 (5)	0.791 (4)
LSID-SISO (ACS)	**0.819 (1)**	**0.814 (1)**	**0.861 (1)**	**0.802 (1)**	**0.824 (1)**
LSID-SISO (ANN)	**0.819 (1)**	0.800 (2)	0.844 (2)	0.796 (3)	0.815 (2)
DeLeS (Graf, Kinshuk et al., 2009)	0.799 (4)	0.790 (6)	0.788 (7)	0.702 (7)	0.770 (7)

Table 48. Comparison of LACC results (ranks in parentheses, top result bolded)

Approach	LACC				
	A/R	S/I	V/V	S/G	Average
LSID-ANN	0.610 (3)	0.575 (3)	0.656 (2)	**0.613 (1)**	0.614 (2)
LSID-ACS	0.599 (4)	0.583 (2)	0.534 (5)	0.426 (6)	0.536 (6)
LSID-GA	0.584 (6)	0.557 (5)	0.541 (4)	0.522 (5)	0.551 (4)
LSID-PSO	0.596 (5)	0.551 (6)	0.482 (6)	0.524 (4)	0.538 (5)
LSID-SISO (ACS)	0.615 (2)	**0.608 (1)**	**0.673 (1)**	0.583 (3)	**0.619 (1)**
LSID-SISO (ANN)	**0.627 (1)**	0.573 (4)	0.638 (3)	0.608 (2)	0.612 (3)
DeLeS (Graf, Kinshuk et al., 2009)	0.435 (7)	0.389 (7)	0.226 (7)	0.134 (7)	0.296 (7)

Table 49. Comparison of %Match results (ranks in parentheses, top result bolded)

Approach	%Match				
	A/R	S/I	V/V	S/G	Average
LSID-ANN	0.986 (4)	0.961 (3)	0.986 (3)	0.986 (2)	0.980 (3)
LSID-ACS	**1.000 (1)**	0.971 (2)	0.909 (5)	0.879 (6)	0.940 (7)
LSID-GA	0.986 (4)	0.946 (5)	0.936 (4)	0.916 (4)	0.946 (6)
LSID-PSO	0.988 (2)	0.971 (2)	0.909 (5)	0.943 (3)	0.953 (5)
LSID-SISO (ACS)	**1.000 (1)**	**1.000 (1)**	**1.000 (1)**	**1.000 (1)**	**1.000 (1)**
LSID-SISO (ANN)	**1.000 (1)**	0.960 (4)	0.986 (3)	**1.000 (1)**	0.987 (2)
DeLeS (Graf, Kinshuk et al., 2009)	0.987 (3)	0.960 (4)	0.987 (2)	0.880 (5)	0.954 (4)

5.2 Working Memory Capacity Identifier

For each WMCID approach, a final result was produced using the optimal parameter and overfitting reduction settings. Overall, the results are straightforward as

WMCID-EANN/R provides the best results in all three metrics (shown in Table 50). Furthermore, it is observed that all of CI algorithms provide an improvement over DeWMC in all metrics.

The improvement by WMCID-EANN and WMCID-EANN/R over the WMCID-ANN (the top mono-CI algorithm approach) shows that the ANN topology is optimizable over the standard 3-layer MLP. Additionally, since the EANN/R provided a better result shows that recurrent connections provide an additional improvement. Although the three ANN variants did best, the optimization algorithms still did well showing that optimizing the weights can provide an improvement to identifying WMC.

The results for the two WMCID-SISO approaches confirm that the justification of using SISO for learning styles is valid. For WMC, there is no observation that multiple solutions are equally good overall but differing for individual students as was observed for learning styles. Correspondingly, there is practically no improvement in the results when using SISO-style architecture for WMC. Some small improvement was expected due to specialization to the smaller data sets; however, the results suggest that this did not occur.

Table 50. Comparison of ACC, LACC and %Match for the WMCID approaches

Approach	ACC	LACC	%Match
WMCID-ANN	0.862 (4)	0.700 (3)	0.907 (2)
WMCID-ACS	0.832 (8)	0.670 (7)	0.876 (4)
WMCID-GA	0.851 (6)	0.694 (4)	0.893 (3)
WMCID-PSO	0.835 (7)	0.685 (5)	0.876 (4)
WMCID-SISO (GA)	0.854 (5)	0.667 (8)	0.893 (3)
WMCID-SISO (ANN)	0.863 (3)	0.673 (6)	0.907 (2)
WMCID-EANN	0.873 (2)	0.708 (2)	**0.952 (1)**
WMCID-EANN/R	**0.880 (1)**	**0.711 (1)**	**0.952 (1)**
DeWMC (Chang et al., 2016)	0.809 (7)	0.442 (7)	0.809 (5)

Chapter VI - Discussion

This chapter aims to discuss the observations made on the LSID and WMCID approaches and is divided into four sections. The four section will discuss observation on the mono-CI algorithm-based LSID approaches, the hybrid LSID approaches, the mono-CI algorithm-based WMCID approaches and the hybrid WMCID approaches in that order.

6.1 Analysis of LSID Mono-CI Algorithms

For the LSID mono-CI algorithms, three analyzes are performed. First, an analysis is done for how the results were achieved, focusing particularly on improvements in precision over DeLeS (Graf, Kinshuk et al., 2009). For each learning style dimension only most precise algorithm is examined meaning LSID-ACS for the A/R and S/I dimensions and LSID-ANN for the V/V and S/G dimensions. The second analysis looks at the distribution and clustering of weight values produced by optimization approach. The third analysis examines the performance of the individual algorithms.

In the A/R dimension, the improvements for LSID-ACS for all three metrics came from a more precise identification of students with a moderate to strong reflective preference ($LS_{actual} \leq 0.313$). Students with +9 or +11 active preference ($\mathrm{LS}_{\mathrm{actual}} \geq 0.929$) are identified worse by LSID-ACS than DeLeS, often identified with a balanced preference ~0.55. Although there are not many of such students (n = 4), it is an area for improvement for LSID-ACS as students with such a strong preference are the ones most penalized by misidentification.

LSID-ACS identifies students with a strong sensing preference ($LS_{actual} \geq 0.929$) and a strong intuitive preference ($LS_{actual} \leq 0.143$) better than DeLeS thereby improving the LACC metric. Despite, the improvement in identifying these students, there is no corresponding improvement in overall precision (ACC) as LSID-ACS is less precise at identifying students with a very balanced preference ($0.438 \geq LS_{actual} \geq 0.563$). LSID-ACS misidentifies these balanced students as having a moderate or strong preference. The worst case is student ID#100 with an S/I value of 0.438, DeLeS identifies this student as 0.461; whereas, LSID-ACS identifies this student as 0.597. This is equivalent to a +1 sensing preference vs the actual +1 intuitive preference. Overall, since students with a balanced preference can handle both sensing and intuitive material this is not such a bad misidentification.

LSID-ANN focuses on correctly identifying visual students as 81.9% of the students in the training data set have a visual preference and this maximizes fitness. This focus can be seen by examining the lowest and average $\mathrm{LS}_{\mathrm{identified}}$ value. The lowest $\mathrm{LS}_{\mathrm{identified}} = 0.51$ and an average of 0.743. By comparison, DeLeS, which has no intelligent mechanism trying to maximize fitness, has a lowest $\mathrm{LS}_{\mathrm{identified}} = 0.08$ and an average of 0.708. The narrower identified value range found by LSID-ANN does improve results as seen with an improvement in both ACC from 0.788 to 0.840. Additionally, although DeLeS find a wider range of identified values, sometimes the more extreme values are very wrong resulting in a low LACC metric (0.226). For example, the student (id#356) with $\mathrm{LS}_{\mathrm{identified}} = 0.08$ has an $\mathrm{LS}_{\mathrm{actual}} = 0.857$ while LSID-ANN identifies this student with an $\mathrm{LS}_{\mathrm{identified}} = 0.750$. Overall though, it would be ideal to have both a wide range of identified values and high accuracy, and as will be discussed in Section 6.2 this was partially resolved with LSID-SISO.

In the S/G dimension, DeLeS struggled to identify many students with a +5 or stronger sequential preference ($LS_{actual} \geq 0.786$) and +5 or stronger global preference ($LS_{actual} \leq 0.313$). In one case, identifying a student with an $LS_{actual} = 0.071$ (+9 global preference) as $LS_{identified} = 0.938$ (approximately a +9 sequential preference), hence the very low LACC value for DeLeS. LSID-ANN improves results by being more precise for students with moderate (± 5) to strong (± 9) preferences. However, LSID-ANN identifies the strongest sequential preferences ($LS_{actual} \geq 0.929$) with a balanced to moderate sequential preference. Similarly, students with the strongest global preferences ($LS_{actual} \leq 0.143$) were identified with a balanced to moderate global preference by LSID-ANN.

Behavior patterns which are the most useful for identifying learning styles should have the highest weights, and vice versa for those behavior patterns which are not as useful. To capture weights for this analysis, the most precise optimization algorithm for each learning style dimension was executed 10 times. For the A/R, S/I and V/V dimensions ACS was used and for the S/G dimension GA was used. Since each execution has 10 folds, this makes for a total of 100 executions for each dimension. The average weight across all executions is calculated for each relevant behavior pattern in each dimension and shown in Tables 51 to 54. The weights exhibited some clustering so, the weights were divided into four regions: 0 to 0.25 (very low), 0.25 to 0.5 (low), 0.5 to 0.75 (high) and 0.75 to 1.0 (very high) and the number of times the weight falls into each region is recorded and shown in Tables 51 to 54. Due to the general inconsistency of the results, it is difficult to draw many conclusions. There is at least one behavior pattern in for each learning style dimension which has a consistently high weight. For example, the "forum_post" and "question_text" behaviors have a consistently high weight for the V/V dimension. Similarly, there is at least one behavior pattern with a consistent low weight for each learning style dimension. To continue the example, the “forum_stay” and “forum_visit” have consistent low weights for the V/V dimension. The analysis suggests that those with consistent high weights are the most useful predictors and those with the consistent low weights are the least useful. This cannot be conclusively stated though because the importance of a weight is relative to the other weight values and too many of the weights are very inconsistent. So although the data hints at possible behavior pattern importance more investigation is required.

Table 51. Average weights per behavior pattern for A/R dimension as found by LSID-ACS

Behavior Pattern	Average	Very Low	Low	High	Very High
content_stay	0.24	54	46	0	0
content_visit	0.59	0	22	67	11
example_stay	0.82	0	0	21	79
execise_stay	0.32	28	65	7	0
exercise_visit	0.51	19	23	43	15
forum_post	0.70	0	28	12	60
forum_visit	0.70	10	0	37	53
outline_stay	0.28	36	64	0	0
quiz_stay_result	0.21	68	32	0	0
self_assess_stay	0.17	85	15	0	0
self_assess_twice_wrong	0.73	0	29	0	71
self_assess_visit	0.46	24	18	58	0

Table 52. Average weights per behavior pattern for V/V dimensions as found by LSID-ACS

Behavior Pattern	Average	Very Low	Low	High	Very High
content_visit	0.53	0	40	58	2
forum_post	0.83	0	0	17	83
forum_stay	0.17	83	17	0	0
forum_visit	0.19	77	23	0	0
question_graphics	0.42	21	56	5	18
question_text	0.77	0	0	40	60

Table 53. Average weights per behavior pattern for S/I dimension as found by LSID-ACS

Behavior Pattern	Average	Very Low	Low	High	Very High
content_stay	0.68	0	23	31	46
content_visit	0.63	9	27	14	50
example_stay	0.22	62	38	0	0
example_visit	0.75	0	3	43	54
exercise_visit	0.42	30	44	26	0
question_concepts	0.63	0	41	28	31
question_detail	0.54	18	22	60	0
question_develop	0.72	2	19	16	63
question_facts	0.36	39	28	31	2
quiz_revision	0.34	39	37	24	0
quiz_stay_result	0.37	37	26	37	0
self_assess_stay	0.67	0	21	42	37
self_assess_visit	0.59	3	24	54	19

Table 54. Average weights per behavior pattern for S/G dimension as found by LSID-GA

Behavior Pattern	Average	Very Low	Low	High	Very High
navigation_overview_stay	0.19	77	23	0	0
navigation_overview_visit	0.85	0	0	9	91
navigation_skip	0.24	54	46	0	0
outline_stay	0.18	78	22	0	0
outline_visit	0.29	39	56	5	0
question_detail	0.46	18	28	54	0
question_develop	0.35	31	44	25	0
question_interpret	0.37	28	47	25	0
question_overview	0.23	65	27	8	0

With respect to overfitting reduction, stratification is clearly the mechanism of choice as it improved results for every algorithm and dimension except LSID-ANN for the S/G dimension. This is not surprising as learning styles distributions tend to be fairly consistent from study to study (Felder & Spurlin, 2005), and this is precisely the sort of situation for which stratification is intended. For LSID-ANN, weight decay was also beneficial by improving results in every dimension. As stratification was useful for the S/G dimension for the remaining LSID approaches, an additional experiment was performed for LSID-ANN with weight decay set to 0.0 (i.e. off) and stratification on. The result was ACC=0.795 which compares favourably to the result with no overfitting reduction at all (ACC=0.792). Thus stratification is useful in all instances; however, for the S/G dimension weight decay provides sufficient overfitting reduction so as to make stratification unnecessary. FEP only provided an improvement for LSID-GA. The drawback to FEP is that it reduces the training set size by 10%, so it is possible with a larger data set that FEP might be more helpful.

Several observations were made on algorithm performance starting with examining the average number of generations completed by the algorithm before terminating (shown in Table 55). First it can be seen that LSID-ACS and LSID-GA had a higher average number of generations than LSID-PSO and LSID-ANN which never exceeded the minimum. For LSID-PSO the low average number of generations is explained by early convergence caused by inefficient trajectories. When the individual and global best positions were close, the particle would tend to orbit them elliptically. When the individual and global best positions are distant, the particles tend towards a flat trajectory between both points. Thus, LSID-PSO quickly stopped finding better solutions and so terminated fairly quickly. Since ANNs operate as a black box (Mitchell, 1997b) it is more difficult to determine why it trained so quickly; however, since LSID-ANN performed best overall of the mono-CI algorithms the quick training did not prevent it from doing well. It may simply be that an ANN is more efficient for this problem.

Table 55. Average number of generations before termination

Algorithm	A/R	S/I	V/V	S/G
LSID-ACS	27745	25311	11307	26907
LSID-GA	32698	27509	10000	27295
LSID-PSO	10000	10000	10000	10000
LSID-ANN	10000	10000	10000	10000

A literature search was done looking for techniques for measuring population diversity in GA, ACS and PSO. Existing techniques for considering diversity in GA either measure the error relative to the global optimum solution (Andre, Siarry, & Dognon, 2001) or are used for multi-objective GA (Farhang-Mehr & Azarm, 2002; Horn, Nafpliotis, & Goldberg, 1994), neither of which are applicable to LSID-GA. Instead diversity was considered in a manner used by Leung et al. (1997) in which they examine how often the same gene value was presented for every genome in the population. This was found to be a very rare occurrence, and never for more than one gene at a time. So it is concluded that diversity was fairly maintained throughout the main part of the processing. A technique found for ranked ant system (Nakamichi & Arita, 2004) in which the number of unique solutions found by the colony each generation are counted is suitable for measuring diversity in ACS since the underlying concepts of the two algorithms are the same. The number of unique solutions was almost always equal to the population size, i.e. very diverse, although occasionally there were a few (5 or less, depending on population size) identical solutions. For LSID-PSO, the technique used to measure diversity has two steps (Riget & Vesterstrøm, 2002). In step 1, after each generation, an *average point* is found by averaging each coordinate in the population. In step 2, diversity is found as the sum of the Euclidian distances for each particle to the average point. Using this technique, it was quite clear that diversity was not well maintained with typical values being in the range from 0.04 to 0.12 × P.

6.2 Analysis of LSID Hybrid Algorithms

As the results for LSID-SISO (ACS) are better than LSID-SISO (ANN) (shown in Tables 43 to 45), except where tied in %Match and LACC for the A/R and S/G dimensions, the remainder of this section will analyze only LSID-SISO (ACS). The analysis will look at: the effects of specialization, the effectiveness of confidence as a splitting mechanism, how results were improved, observations on optimal parameters overfitting reduction settings and algorithm performance issues. In examining how resulted are improved, LSID-SISO (ACS) is compared to the most precise mono-CI algorithm, i.e. LSID-ACS for the A/R and S/I dimensions and LSID-ANN for the V/V and S/G dimensions.

LSID-SISO (ACS) is expected to provide improvements by two means: by capitalizing on the benefits of multiple equally good solutions, in essence allowing HICON and LOWCON to specialize to their data set. To show evidence of specialization the following calculation is made. For each student, the ACC result from the best mono-CI approach is subtracted from the ACC result from LSID-SISO (ACS) giving a ΔACC for that student. So, if ΔACC is positive than LSID-SISO (ACS) improved the precision of identification for the student and vice versa. The ΔACC are divided into two groups based on whether the student was identified by HICON or LOWCON. Finally the ΔACC for HICON and LOWCON are summed separately. If specialization is occurring then it would be expected that the sum of the ΔACC values (ΣΔACC) should be positive for both HICON and LOWCON and this can be seen in Table 56.

Table 56. ACC improvement by dimension for HICON and LOWCON

ΣΔACC	HICON	LOWCON
A/R	+0.004	+0.022
S/I	+0.054	+0.015
V/V	+0.035	+0.012
S/G	+0.009	+0.028

Overall using confidence as a means to separate the students works well at providing an improvement in the performance metrics; however, a deeper analysis shows that it has mixed results for getting each student sent to the proper solving algorithm

(HICON or LOWCON). A student is considered sent to the proper solver if the initial prediction ($LS_{predicted}$) value is within ±0.25 of the LS_{actual} and the confidence ≥ 0.75 or if the $LS_{predicted}$ value is not within ±0.25 of the LS_{actual} and the confidence < 0.75. Table 57 shows the percentage of students at each solver that were placed correctly. The results show that overall the algorithm is not picking by guessing as otherwise the percentages would be closer to 50%.

Table 57. Percentage of students sent to correct solver

	HICON %Correct	LOWCON % Correct
A/R	72%	84%
S/I	83%	74%
V/V	85%	73%
S/G	73%	80%

For the A/R and S/G dimensions, identifying students with a poor initial prediction by expressing a low confidence in the result worked well with 84% and 80% identified properly. The inverse is true for the S/I and V/V dimensions, where students with a good initial prediction had a high confidence value 83% and 85% respectively. Except for the V/V dimension, students who ended up at the wrong solving algorithm were students with a balanced preference for that dimension more than 90% of the time indicating that it might be more difficult to have high confidence in the identification for students with a balanced preference. However, since balanced students can more easily handle learning material for both preferences, a lack of confidence in their identification is not as potentially harmful. In any case, balanced students were not badly misidentified despite being sent to the wrong solver, except for the slight loss of precision for balanced students in the A/R dimension.

On examining the students on an individual basis in the A/R dimension, it is observed that when compared to LSID-ACS students are identified slightly better on average, while students with a balanced preference are identified slightly worse. Overall, this results in a small increase for the LACC metric without a corresponding increase in precision. By identifying students with stronger preferences more precisely is a practical improvement as students with a balanced preference are more able to adapt to any material than those with a strong preference.

In the S/I dimension, the increase in precision is obtained as a general increase for all students as no student was identified worse by LSID-SISO (ACS) than LSID-ACS. The improvement in precision caused the increase in LACC and %Match metrics.

Table 58. Comparison of ACC values between LSID-ANN and LSID-SISO (ACS)

Student ID	LS_{actual}	LS_{id} LSID-ANN	ACC LSID-ANN	LS_{id} LSID-SISO (ACS)	ACC LSID-SISO (ACS)	ΔACC
75	0.438	0.726	0.712	0.648	0.790	+0.078
175	0.438	0.805	0.633	0.683	0.755	+0.122
200	0.438	0.691	0.747	0.583	0.855	+0.108
242	0.438	0.837	0.601	0.621	0.817	+0.216
295	0.438	0.831	0.607	0.630	0.808	+0.201
593	0.438	0.759	0.679	0.675	0.763	+0.084
129	0.313	0.693	0.620	0.589	0.724	+0.104
177	0.313	0.690	0.623	0.590	0.723	+0.100
225	0.313	0.704	0.609	0.515	0.798	+0.189
72	0.214	0.756	0.458	0.590	0.624	+0.166
255	0.214	0.687	0.527	0.335	0.879	+0.352

The improvement by LSID-SISO (ACS) in the V/V dimension is obtained almost entirely by improving the results for the verbal students. For each verbal student a ΔACC value is calculated by subtracting LSID-ANN's ACC values from the ACC value for LSID-SISO (ACS). Table 58 shows the calculated ΔACC, the ACC values, the actual (*LS_{actual}*) and identified learning style (*LS_{id}*) values from LSID-ANN and LSID-SISO (ACS). For each student, the ACC was improved by LSID-SISO (ACS) in a range from

0.078 to 0.352. However, it can be seen that despite the improvement all but one verbal student (student #255) are identified as having a visual preference (even if it is slight).

For the S/G dimension, the increase in precision when compared to LSID-ANN is fairly small and split among many students. In six of the folds a single student had a drop in precision, hence the drop in LACC.

As with the mono-CI algorithm-based approaches, an analysis of algorithm performance begins by examining the average number of generations before termination. Table 59 shows the average for each step in the LISD-SISO (ACS) architecture. For the *Prediction* step with ACS, the number of generations is similar to LSID-ACS and this is expected as the *Prediction* step ACS is identical to LSID-ACS. For steps which use an ANN, the algorithm never requires more than the minimum of 10,000 generations. Diversity measurements for the prediction step were performed by counting the number of unique solutions (Nakamichi & Arita, 2004). As with LSID-ACS, the number of unique solution was almost always equal to the population size, and again this is expected as they are the same algorithm. As with LSID-ANN, it is difficult to assess the inner workings of an ANN since they operate as a black box (Mitchell, 1997b); however, based on the overall results for LSID-SISO (ACS) they seemed to work well.

Table 59. Average number of generations before termination

Algorithm	A/R	S/I	V/V	S/G
Prediction (ACS)	28312	28501	10508	24302
Confidence	10000	10000	10000	10000
HICON	10000	10000	10000	10000
LOWCON	10000	10000	10000	10000

6.3 Analysis of WMCID Mono-CI Algorithms

The same three analyzes are performed for the WMCID mono-CI algorithms as for the LSID algorithms. First, an analysis is done for how the results were achieved, focusing particularly on improvements in precision of WMCID-ANN (the best approach) over DeWMC (Chang et al., 2013). The second analysis looks at the distribution and clustering of weight values produced by optimization approach. The third analysis examines the overfitting setting and performance of the individual algorithms.

By conducting a closer examination of the results for each individual student, some additional observations are made. WMCID-ANN improved the identification accuracy (ACC) for every individual student with a WMC higher than 0.3 compared to DeWMC. For students with a WMC between 0.3 and 0.7 (63.4% of students in the dataset), WMCID-ANN has an average ACC of 0.914. When using WMCID-ANN, the average ACC for students with a WMC higher than 0.7 (28.6% of students in the dataset) is 0.791. However, the average ACC for students with a WMC lower than 0.3 (8.0% of students in the dataset) is 0.705. In contrast, for DeWMC, the average ACC for students with a WMC higher than 0.7 is 0.684 and the average ACC for students with a WMC lower than 0.3 is 0.748. As can be seen from these values, WMCID-ANN is identifying students with moderate and high WMC better than DeWMC but worse for those with low WMC. Most likely, this is caused by the ANN not having enough data for students with very high and especially very low WMC and a larger sample size would help improving the results of the algorithm even further.

Since no weight clustering was observed for WMC a slightly different weight analysis was performed than for learning styles. For each pattern, the minimum, maximum and average weights across all folds from the final result are shown in Table 60. Furthermore, Table 60 shows the percentage of learning sessions in which each pattern was activated. From this table, it can be seen that the linear navigation, recalling learned material and learning styles patterns are weighted rather low, suggesting that the impact of these patterns on identifying WMC is smaller than the impact of other patterns. The weights for the constant reverse navigation and performing simultaneous tasks patterns are generally high, indicating that they are highly predictive of WMC. However, the performing simultaneous tasks pattern appears only in 8.25% of all learning sessions of all students. Accordingly, its role in identifying WMC should be investigated further with a sample where this pattern occurs more often.

Table 60. Minimum, Maximum, and Average Weights and Percentage of Activated Learning Sessions per Pattern

Pattern	Min	Max	Average	Activated
Linear Navigation	0.03	0.13	0.07	89.98%
Constant Reverse Navigation	0.50	0.99	0.82	78.62%
Performing Simultaneous Tasks	0.81	1.00	0.97	8.25%
Recalling Learned Material	0.10	0.33	0.22	58.86%
Revisiting Passed Learning Objects	0.36	0.84	0.62	60.19%
Learning Styles	0.02	0.17	0.10	100.00%

To analyse algorithm performance, first the average number of generations completed by the algorithm before terminating (shown in Table 61) was examined. WMCID-ACS and WMCID-GA both required more than the minimum number of generations, but only for 3 and 4 folds out of 10 respectively. WMCID-PSO and WMCID-ANN both never required more than the minimum.

Table 61. Average number of generations before termination

Algorithm	Avg # of Generations
WMCID-ACS	11795
WMCID-GA	15327
WMCID-PSO	10000
WMCID-ANN	10000

The next analysis of performance was to examine population diversity for the optimization algorithms. For WMCID-GA, the number of unique values was examined on a gene by gene basis in a manner similar to Leung et al. (1997) to ensure that diversity was being maintained. In general, gene value diversity was very high until WMCID-GA approached convergence, which was somewhat different than LSID-GA. This could be because for WMC, unlike learning styles there are not multiple equally good solutions, thus the GA was converging towards the single global optimal solution. Counting the number of unique solutions found by the ants per generation (Nakamichi & Arita, 2004) to measure for WMCID-ACS showed that number of unique solutions was almost always equal to the population size. As with WMCID-GA, diversity did drop off slightly as the algorithm reached convergence although not to the same degree. WMCID-PSO has similar problems with diversity, measured as the sum of the Euclidian distances to an average point (Riget & Vesterstrøm, 2002), and inefficient trajectories as LSID-PSO.

With respect to overfitting reduction, when iteratively evaluating stratification and FEP it was observed that for every algorithm when each overfitting reduction technique was used alone it would improve results. As seen in section 4.5, the best improvement was found when they were used together.

6.4 Analysis of WMCID Hybrid Algorithms

As WMCID-EANN/R had better overall results that WMCID-EANN, the discussion on improvements will focus mainly on the recurrent version. The improvement in results over WMCID-ANN comes as a result of improving the identification of students with low WMC. With WMCID-ANN, students with WMC <

0.4 were identified with a WMC between 0.4 and 0.5. WMCID-EANN/R identifies these students with greater precision and so this improves all three metrics. As was done for WMCID-ANN, the average ACC was calculated for student in the ranges of WMC > 0.7, WMC between 0.3 and 0.7 and WMC < 0.3. For students with a WMC between 0.3 and 0.7 (63.4% of students in the dataset), WMCID-EANN/R has an average ACC of 0.918. The average ACC for students with a WMC higher than 0.7 (28.6% of students in the dataset) is 0.838. However, the average ACC for students with a WMC lower than 0.3 (8.0% of students in the dataset) is 0.732. In contrast, for WMCID-ANN, the average ACC for students between 0.3 and 0.7 is 0.914, the average with a WMC higher than 0.7 is 0.791, the average ACC for students with a WMC lower than 0.3 is 0.705. There is not much improvement for students with WMC higher than 0.7; however, they were already well identified. The greatest improvement is seen for students with WMC between 0.3 and 0.7 and considerable improvement for students with WMC lower than 0.3. Despite the improvement for students with WMC < 0.3, WMCID-EANN/R is still a little bit worse than DeWMC which has an average WMC of 0.748 for those students.

Since recurrent links improved results an analysis was made of the nature of the links. Table 62 shows the minimum, maximum and average number of links which were recurrent either to the same layer or to a previous layer. In addition, the number of links which were to and from the same node (links to self) is indicated in parentheses. It can be seen that the number and distribution of links is fairly consistent, with many input-to-input links a few hidden-to-input links and never any hidden-to-hidden links. Also, the number of links to self is generally quite low.

Table 62. Average number of recurrent links for WMCID-EANN/R (number of links to self in parenthesis)

Link Type	# of Links		
	Min	Max	Avg
From Input to Input	63 (4)	96 (10)	78 (7)
From Hidden to Input	4 (n/a)	10 (n/a)	8 (n/a)
From Hidden to Hidden	0 (0)	0 (0)	0 (0)

In terms of algorithm performance, the average number of generations prior to termination was recorded (shown in Table 63). It can be seen that it takes more generations to complete training both the EANN and EANN/R than for WMCID-ANN. Additionally, each generation requires much more processing time since each genome must be decoded into an ANN and then the ANN must be trained and evaluated. Thus overall the processing time is greatly increased with the EANN and EANN/R approaches averaging ~6 minutes on an i7-4770 @ 3.40 GHz. for a single fold (execution). By comparison, WMCID-ANN completes a single fold in < 5 seconds on the same computer. For the GA part of the EANN and EANN/R, gene diversity was checked by examining the number of times the same gene value appears in every genome in the population (Leung et al., 1997). As with other uses of GA, this proved to be extremely rare even as the GA approached convergence.

Table 63. Average number of generations before termination

Algorithm	Avg # of Generations
WMCID-ANN	10000
WMCID-EANN	13716
WMCID-EANN/R	15101

Chapter VII - Conclusions

The overarching focus of this research was to investigate how CI algorithms could be used to identify learning styles and WMC from student behaviors when using LMSs. Other related works develop an entirely new behavior framework of unknown quality based on literature; however, for this research it was decided to instead pick a leading approach and try to improve it with CI algorithms. By using an existing leading approach, it was possible to know that the behavior patterns were at least already somewhat effective for identifying learning style or WMC. A literature review found “Detecting Learning Styles” (DeLeS) (Graf, Kinshuk et al., 2009) and “Detecting Working Memory Capacity” (DeWMC) (Chang et al., 2013). DeLeS has an accuracy between 73% and 79% based on the SIM metric which is equal to or greater than other related works (ignoring those with simulated data or major limitations); thereby, making it a leading approach. DeWMC has an accuracy of 81% based on absolute error and since no other automatic approach could be found in literature, it was treated as the de facto leader.

The plan to improve DeLeS and DeWMC occurred in two broad phases. First, an investigation was done on using mono-CI algorithms to improve DeLeS and DeWMC. Second, the results from the first phase would be analyzed and an appropriate hybrid CI algorithm was developed based on any observations made from the analysis. The first phase was divided into two approaches: classification and optimization. For the classification approach, an ANN was used as it can find complex functions to classify data. DeLeS and DeWMC use an unweighted average of hint values generated from behavior data. So the optimization approach worked by finding an optimal set of pattern weights. Since little was known of the solution space describing the weights, it was decided that three different optimization algorithms should be used as each uses a different search mechanism which is ideal for different solution spaces. The three optimization algorithms selected were ant colony system, genetic algorithm and particle swarm optimization. For learning styles, these approaches were called LSID-ANN, LSID-ACS, LSID-GA and LSID-PSO while for WMC they were called WMCID-ANN, WMCID-ACS, WMCID-GA and WMCID-PSO. To evaluate the performance of the approaches, three metrics were used for WMC and four for learning styles. Accuracy (ACC) measured overall performance while lowest ACC (LACC) and the percentage of students matched with reasonable accuracy (%Match) were used to measure performance for individual students. The fourth metric for learning styles was similarity (SIM) which is used commonly in literature to measure overall performance identification (García et al., 2007; Graf, Kinshuk et al., 2009; Özpolat & Akar, 2009).

The result from the first phase showed that for learning styles the best approach was split between LSID-ACS, being best for the A/R and S/I dimensions, and LSID-ANN being best for the V/V and S/G dimensions. When compared to DeLeS, LSID-ACS provided an improvement in precision for each dimension; thereby, proving that finding an optimal set of weights is a valid approach. LSID-ANN provided an improvement over DeLeS in precision for each dimension except S/I, where it did improve the LACC metric. LSID-ANN also was more precise than LSID-ACS, LSID-GA and LSID-PSO in the V/V and S/G dimensions; therefore, this proves that an ANN is able to find a better function for identifying learning styles than averaging hint values (as used by DeLeS), even when the weights are optimized. Overall, when considering the average across all dimension, LSID-ANN had the best results.

For the second phase, it was observed that the LSID approaches would have similar overall results across multiple executions while different individual students would be identified better or worse. This suggested that performance could be improved by splitting the students into optimal sub-groups and then identify them with an ANN specialized to the sub-groups. This was accomplished building an approach called LSID-SISO (Simply and Solve) based on a loosely coupled hybrid architecture. The architecture consisted of three steps and four algorithms. The first step was the *Prediction* step, in which either an LSID-ANN or LSID-ACS was used to provide an initial identification of the learning style making two versions of LSID-SISO, called LSID-SISO (ACS) and LSID-SISO (ANN). The second step was the *Simplify* step, where an ANN was used to produce a confidence value on the initial predicted identification. Based on the confidence value, the students were split into high and low confidence groups. Each of the groups was then sent to a separate ANN that provides the final identification of the learning styles. LSID-SISO (ACS) was found to have better results than LSID-SISO (ANN). LSID-SISO (ACS) was found to improve results in most metrics when compared to the best mono-CI approach for each learning style dimension, i.e. LSID-ACS for the A/R and S/I dimensions and LSID-ANN for the V/V and S/G dimensions. LSID-SISO (ACS) tied LSID-ACS in ACC for the A/R dimension and had a lower LACC than LSID-ANN for the S/G dimension).

The results for the first phase for WMC show that every mono-CI algorithm provided an improvement over DeWMC in every metric. Thus, unquestionably, both optimizing the weights for DeWMC and using the behavior patterns and learning styles as inputs to an ANN were effective at providing an improvement to precision and fairness. Overall, WMCID-ANN is the best approach in all metrics, with WMCID-GA as the best of the optimization algorithms.

Since the ANN was the best mono-CI algorithm for identifying WMC, for the second phase, it was decided to use the evolving ANN (EANN) as the hybrid algorithm. One drawback to the ANN is that it uses a fixed topology which may be non-optimal and the EANN resolves this by using an evolutionary algorithm, such as GA, to search for an optimal topology. EANN can produce recurrent and non-recurrent topologies and so both were evaluated using a hybrid training model. The recurrent artificial neural network topology was found to provide an improvement in every metric over WMCID-ANN although it did have a longer training time requirement.

The benefit of this research lay in supporting learning through student modelling. A student model is a queryable collection of information about students including elements such as learning styles and WMC (Brusilovsky & Millán, 2007). Student models support teachers by providing them with better insight into their students' profiles. This allows the teachers to offer more appropriate interventions, especially when a student struggles (Delozanne, Grugeon, Previt, & Jacoboni, 2003; Graf, Kinshuk et al., 2009; Lin, 2004). Students are empowered by knowing about themselves (Felder & Spurlin, 2005) and benefit by understanding their strengths and weakness with respect to learning styles and WMC and so make better self-regulated learning choices. In addition to providing information directly to teachers and students, the student models are used by adaptive learning systems to optimize the learning environment to each student’s preferences and abilities (Brusilovsky & Millán, 2012). A more precise student model, in the case of this research with respect to learning styles and WMC, allows for adaptations to be more closely matched to each student. Since providing such adaptations based on learning styles and WMC has been shown to improve learning outcomes (Bajraktarevic et al., 2003; Paas et al., 2004), satisfaction (Cordova & Lepper, 1996; Popescu, 2010), learning transfer (Moreno, 2004; Van Merriënboer et al., 2002) and reduce the time needed to learn (Cooper, 1998; Graf et al., 2009), it follows that the more precise identification of learning styles and WMC provided by the LSID and WMCID approaches will allow students to learn better and faster from courses using adaptive learning systems.

The future of this research rests in three possible directions. One potential avenue is to use feature selection techniques to consider turning the behavior patterns on / off for each learning styles dimension thereby allowing the remaining behavior patterns to have better weights. Secondly, adaptive mechanisms, such as a decreasing inertia for LSID-PSO, could be evaluated to see if they can further improve precision. Finally, although the data set is sufficient for showing that CI algorithms can improve precision of learning styles identification, LSID should be evaluated with a more diverse and larger data set. A diverse data set should show that LSID works also with students from different educational levels (i.e. primary school, secondary school, etc.), from various fields of study and backgrounds.

REFERENCES

Abido, M. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems, 24*(7), 563-571.

Aghdam, M. H., Ghasem-Aghaee, N., & Basiri, M. E. (2009). Text feature selection using ant colony optimization. *Expert Systems with Applications, 36*(3), 6843-6853.

Andre, J., Siarry, P., & Dognon, T. (2001). An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. *Advances in engineering software, 32*(1), 49-60.

Ayres, P. L. (1993). Why goal-free problems can facilitate learning. *Contemporary Educational Psychology, 18*(3), 376-381.

Azevedo, R., Cromley, J. G., Winters, F. I., Moos, D. C., & Greene, J. A. (2005). Adaptive human scaffolding facilitates adolescents' self-regulated learning with hypermedia. *Instructional science, 33*(5-6), 381-412.

Bacon, D. R. (2004). An examination of two learning style measures and their association with business learning. *Journal of Education for Business, 79*(4), 205-208.

Baddeley, A. (1992). Working memory. *Science, 255*(5044), 556-559.

Bajraktarevic, N., Hall, W., & Fullick, P. (2003, June 22). *Incorporating learning styles in hypermedia environment: Empirical evaluation.* Paper presented at the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, Johnstown, Pennsylvania.

Barbe, W. B., & Milone Jr, M. N. (1981). What We Know about Modality Strengths. *Educational Leadership, 38*(5), 378-380.

Barbe, W. B., Swassing, R. H., & Milone Jr, M. N. (1979). *Teaching through modality strengths: Concepts and practices*. New York, NY: Zaner-Bloser.

Beaumont, I. H. (1994). User modelling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction, 4*(1), 21-45.

Beck, J. E., & Chang, K.-m. (2007). Identifiability: A fundamental problem of student modeling *User Modeling 2007* (pp. 137-146): Springer.

Beebe-Center, J. G., Rogers, M., & O'connell, D. (1955). Transmission of information about sucrose and saline solutions through the sense of taste. *The Journal of Psychology, 39*(1), 157-160.

Beilock, S. L., & Carr, T. H. (2005). When high-powered people fail working memory and "choking under pressure" in math. *Psychological Science, 16*(2), 101-105.

Belew, R. K., McInerney, J., & Schraudolph, N. N. (1990). *Evolving networks: Using the genetic algorithm with connectionist learning*. Paper presented at the 2nd Conference on Artificial Life.

Boyle, C., & Encarnacion, A. O. (1998). MetaDoc: an adaptive hypertext reading system *Adaptive Hypertext and Hypermedia* (pp. 71-89): Springer.

Brajnik, G., Guida, G., & Tasso, C. (1987). User modeling in intelligent information retrieval. *Information Processing & Management, 23*(4), 305-320.

Brownell, W. A. (1933). On the accuracy with which reliability may be measured by correlating test halves. *The Journal of Experimental Education, 1*(3), 204-215.

Brusilovsky, P. (1992). *A framework for intelligent knowledge sequencing and task sequencing*. Paper presented at the 2nd International Conference on Intelligent Tutoring Systems.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction, 6*(2-3), 87-129.

Brusilovsky, P. (2012). Adaptive hypermedia for education and training. *Adaptive technologies for training and education, 46*.

Brusilovsky, P., & Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web: Methods and Strategies of Web Personalization* (pp. 3-53). Berlin, Heidelberg: Springer Berlin Heidelberg.

Bull, S. (1998). *'Do It Yourself' Student Models for Collaborative Student Modelling and Peer Interaction.* Paper presented at the 4th International Conference on Intelligent Tutoring Systems, San Antonio, Texas.

Carmona, C., Castillo, G., & Millán, E. (2008, July). *Designing a dynamic bayesian network for modeling students' learning styles.* Paper presented at the Eighth IEEE International Conference on Advanced Learning Technologies.

Carver, C. A., Jr, Howard, R. A., & Lane, W. D. (1999). Addressing different learning styles through course hypermedia. *IEEE Transactions on Education, 42*(1), 33-38.

Cha, H. J., Kim, Y. S., Park, S. H., Yoon, T. B., Jung, Y. M., & Lee, J.-H. (2006, June 26-30). *Learning Styles Diagnosis Based on User Interface Behaviors for the Customization of Learning Interfaces in an Intelligent Tutoring System.* Paper presented at the 8th International Conference on Intelligent Tutoring Systems, Berlin.

Chandler, P., & Sweller, J. (1992). The split-attention effect as a factor in the design of instruction. *British Journal of Educational Psychology, 62*(2), 233-246.

Chang, T.-W., El-Bishouty, M. M., Graf, S., & Kinshuk. (2013). *An Approach for Detecting Students' Working Memory Capacity from Their Behavior in Learning Systems.* Paper presented at the 13th International Conference on Advanced Learning Technologies.

Chang, T.-W., El-Bishouty, M. M., Kinshuk, & Graf, S. (2016). *Identifying Students' Working Memory Capacity in Learning Systems*. Technical Report.

Chen, P.-M., & Kuo, F.-C. (2000). An information retrieval system based on a user profile. *Journal of Systems and Software, 54*(1), 3-8.

Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications, 40*(11), 4715-4729.

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58-73.

Coffield, F., Moseley, D., Hall, E., & Ecclestone, K. (2004). *Learning styles and pedagogy in post-16 learning: a systematic and critical review*. London: Learning & Skills Research Centre.

Conway, A. R., Cowan, N., Bunting, M. F., Therriault, D. J., & Minkoff, S. R. (2002). A latent variable analysis of working memory capacity, short-term memory capacity, processing speed, and general fluid intelligence. *Intelligence, 30*(2), 163-183.

Cook, D. A., & Smith, A. J. (2006). Validity of index of learning styles scores: multitrait– multimethod comparison with three cognitive/learning style instruments. *Medical education, 40*(9), 900-907.

Cooper, G. (1998). Research into cognitive load theory and instructional design at UNSW. Retrieved from http://dwb4.unl.edu/Diss/Cooper/UNSW.htm

Cordova, D. I., & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology, 88*(4), 715.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *psychometrika, 16*(3), 297-334.

Cronbach, L. J., & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*(4), 281.

Dahlstrom, E., Brooks, D. C., & Bichsel, J. (2014). *The Current Ecosystem of Learning Management Systems in Higher Education: Student, Faculty, and IT Perspectives*. Louisville, CO: Educause Center for Analysis and Research.

Daneman, M., & Carpenter, P. A. (1980). Individual differences in working memory and reading. *Journal of Verbal Learning and Verbal Behavior, 19*(4), 450-466.

Daneman, M., & Merikle, P. M. (1996). Working memory and language comprehension: A meta-analysis. *Psychonomic bulletin & review, 3*(4), 422-433.

De Neys, W., d Ydewalle, G., Schaeken, W., & Vos, G. (2002). A Dutch, computerized, and group administrable adaptation of the operation span test. *Psychologica Belgica, 42*(3), 177-190.

DeCaro, R., Peelle, J. E., Grossman, M., & Wingfield, A. (2016). The two sides of sensory-cognitive interactions: effects of age, hearing acuity, and working memory span on sentence comprehension. *Frontiers in psychology, 7*, 236.

Delozanne, E., Grugeon, B., Previt, D., & Jacoboni, P. (2003). *Supporting teachers when diagnosing their students in algebra.* Paper presented at the Artificial Intelligence in Education, Sydney, Australia.

Dorça, F. A., Lima, L. V., Fernandes, M. A., & Lopes, C. R. (2013). Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Expert Systems with Applications, 40*(6), 2092-2101. doi:10.1016/j.eswa.2012.10.014

Dorigo, M., & Gambardella, L. M. (1997a). Ant colonies for the travelling salesman problem. *BioSystems, 43*(2), 73-81.

Dorigo, M., & Gambardella, L. M. (1997b). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*(1), 53-66.

Dorigo, M., & Stützle, T. (2010). Ant Colony Optimization: Overview and Recent Advances. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 227-263). Boston, MA: Springer US.

Dunn, R., DeBello, T., Brennan, P., Krimsky, J., & Murrain, P. (1981). Learning style researchers define differences differently. *Educational Leadership, 38*(5), 372-375.

Eberhart, R. C., & Kennedy, J. (1995, October 4-6). *A new optimizer using particle swarm theory.* Paper presented at the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan.

Encarnação, L. M. (1997). *Multi-level user support through adaptive hypermedia: a highly application-independent help component.* Paper presented at the 2nd International Conference on Intelligent User Interfaces.

Engle, R. W. (2010). Role of working-memory capacity in cognitive control. *Current anthropology, 51*(S1), S17-S26.

Engle, R. W., Tuholski, S. W., Laughlin, J. E., & Conway, A. R. (1999). Working memory, short-term memory, and general fluid intelligence: a latent-variable approach. *Journal of Experimental Psychology: General, 128*(3), 309.

Ericsson, M., Resende, M. G. C., & Pardalos, P. M. (2002). A genetic algorithm for the weight setting problem in OSPF routing. *Journal of combinatorial optimization, 6*(3), 299-333.

Farhang-Mehr, A., & Azarm, S. (2002, May 12-17). *Diversity assessment of Pareto optimal solution sets: an entropy approach.* Paper presented at the World Congress on Computational Intelligence, Honolulu, Hawaii.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education, 78*(7), 674-681.

Felder, R. M., & Soloman, B. A. (2000). Learning styles and strategies. Retrieved from https://www.dal.ca/content/dam/dalhousie/pdf/management/Faculty%20%26%20Staff/LEARNING%20STYLES%20AND%20STRATEGIES.pdf

Felder, R. M., & Solomon, B. A. (1998). Index of learning styles Retrieved from http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Felder, R. M., & Spurlin, J. (2005). Applications, reliability and validity of the index of learning styles. *International Journal of Engineering Education, 21*(1), 103-112.

Fleming, N. D. (1995). *I'm different; not dumb. Modes of presentation (VARK) in the tertiary classroom.* Paper presented at the Research and Development in Higher Education.

Frias-Martinez, E., Chen, S. Y., & Liu, T.-C. (2007). Automatic cognitive style identification of digital library users for personalization. *Journal of the American Society for Information Science and Technology, 58*(2), 237-251.

Gambardella, L. M., Taillard, E., & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *Journal of the operational research society*, 167-176.

García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education, 49*(3), 794-808.

Garner, W. (1953). An informational analysis of absolute judgments of loudness. *Journal of experimental psychology, 46*(5), 373.

Genovese, J. E. (2004). The Index of Learning Styles: An Investigation of its Reliability and Concurrent Validity with the Preference Test. *Individual Differences Research, 2*(3).

Gohar, A., Adams, A., Gertner, E., Sackett-Lundeen, L., Heitz, R., Engle, R., Haus, E., & Bijwadia, J. (2009). Working memory capacity is decreased in sleep-deprived internal medicine residents. *Journal of clinical sleep medicine, 5*(3), 191.

Gonçalves, J. F., de Magalhães Mendes, J. J., & Resende, M. c. G. (2005). A hybrid genetic algorithm for the job shop scheduling problem. *European journal of operational research, 167*(1), 77-95.

Graf, S. (2007). *Adaptivity in learning management systems focussing on learning styles.* Vienna University of Technology.

Graf, S., Chung, H. L., Liu, T.-C., & Kinshuk. (2009, July). *Investigations about the effects and effectiveness of adaptivity for students with different learning styles.* Paper presented at the 9th IEEE International Conference on Advanced Learning Technologies, Latvia.

Graf, S., Kinshuk, & Liu, T.-C. (2009). Supporting teachers in identifying students' learning styles in learning management systems: An automatic student modelling approach. *Educational Technology & Society, 12*(4), 3–14.

Graf, S., Lin, T., & Kinshuk. (2007). *Analysing the relationship between learning styles and cognitive traits.* Paper presented at the 7th IEEE International Conference on Advanced Learning Technologies.

Grefenstette, J. J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics, 16*(1), 122-128.

Hammond-Kaarremaa, L. (1994). Rethinking university teaching: a framework for the effective use of educational technologies by Diana Laurillard. Routledge, 1993. *Open praxis: the bulletin of the International Council for Distance Education*(2), 32.

Hayes, J. (1952). Memory span for several vocabularies as a function of vocabulary size. *Quarterly Progress Report*, 338-352.

Heise, D. R. (1969). Separating reliability and stability in test-retest correlation. *American sociological review*, 93-101.

Honey, P., & Mumford, A. (1992). *The manual of learning styles* (3rd ed.): Maidenhead.

Honey, P., & Mumford, A. (2006). *The learning styles questionnaire: 80-item version*: Peter Honey Publications Limited.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). *A niched Pareto genetic algorithm for multiobjective optimization.* Paper presented at the World Congress on Computational Intelligence.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.

Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks, 3*(5), 551-560.

Huai, H. (2000). Cognitive style and memory capacity: effects of concept mapping as a learning method. *Unpublished doctoral dissertation, University of Twente, Enschede: The Netherlands*.

Huang, S.-J. (2001). Enhancement of hydroelectric generation scheduling using ant colony system based optimization approaches. *Energy Conversion, IEEE Transactions on, 16*(3), 296-301.

Jayaprakash, S. M., Moody, E. W., Lauría, E. J., Regan, J. R., & Baron, J. D. (2014). Early alert of academically at-risk students: An open source analytics initiative. *Journal of Learning Analytics, 1*(1), 6-47.

Jia, B., Zhong, S., Wang, W., & Yang, B. (2009). The construction and evolution of learner model in adaptive learning system. 148-152.

Jung, C. G. (1971). *Psychological types*. Princeton, N.J.: Princeton University Press.

Jurafsky, D., Wooters, C., Tajchman, G., Segal, J., Stolcke, A., Foster, E., & Morgan, N. (1994, Sept 18-22). *The Berkeley Restaurant Project.* Paper presented at the 3rd International Conference on Spoken Language Processing, Yokohama, Japan.

Kane, M. J., & Engle, R. W. (2000). Working-memory capacity, proactive interference, and divided attention: limits on long-term memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*(2), 336.

Kao, Y.-T., & Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied Soft Computing, 8*(2), 849-857.

Kass, R. (1989). Student modeling in intelligent tutoring systems—implications for user modeling *User models in dialog systems* (pp. 386-410): Springer.

Keefe, J. W. (1979). Learning style: An overview. *Student learning styles: Diagnosing and prescribing programs, 1*, 1-17.

Kirschner, P. A. (2002). Cognitive load theory: Implications of cognitive load theory on the design of learning. *Learning and instruction, 12*(1), 1-10.

Klašnja-Milićević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). E-Learning personalization based on hybrid recommendation strategy and learning style identification. *Computers & Education, 56*(3), 885-899.

Klein, K., & Fiss, W. H. (1999). The reliability and stability of the Turner and Engle working memory task. *Behavior Research Methods, Instruments, & Computers, 31*(3), 429-432.

Kline, P. (2013). *Handbook of psychological testing*: Routledge.

Kohavi, R. (1995, August). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Paper presented at the International Joint Conference on Artificial Intelligence, Montreal, Quebec.

Kolb, D. A. (1971). *Individual learning styles and the learning process*. Cambridge, MA: Sloan School of Management, Massachusetts Institute of Technology.

Kolb, D. A., & Hay, T. (1999). *Learning style inventory: Version 3*: Hay/McBer Training Resources Group Boston, MA.

Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems, 4*, 950-957.

Kuljis, J., & Liu, F. (2005, July 17-19). *A comparison of learning style theories on the suitability for eLearning*. Paper presented at the Web Technologies, Applications, and Services, Calgary, AB.

Latham, A., Crockett, K., McLean, D., & Edmonds, B. (2012). A conversational intelligent tutoring system to automatically predict learning styles. *Computers & Education, 59*(1), 95-109.

Leung, Y., Gao, Y., & Xu, Z.-B. (1997). Degree of population diversity-a perspective on premature convergence in genetic algorithms and its markov chain analysis. *IEEE Transactions on Neural Networks, 8*(5), 1165-1176.

Ley, T., Kump, B., & Gerdenitsch, C. (2010). Scaffolding self-directed learning with personalized learning goal recommendations *User modeling, adaptation, and personalization* (pp. 75-86): Springer.

Limongelli, C., Sciarrone, F., Temperini, M., & Vaste, G. (2009). Adaptive learning with the LS-plan system: a field evaluation. *Learning Technologies, IEEE Transactions on, 2*(3), 203-215.

Lin, P. (2004). *Supporting teachers on designing problem-posing tasks as a tool of assessment to understand students' mathematical learning.* Paper presented at the 28th Annual Meeting of the International Group for the Psychology of Mathematics Education.

Lin, T. (2007). *Cognitive Trait Model for adaptive learning environments.* (Philosophy in Information System), Massey University, Palmerston North, New Zealand.

Lin, T., Kinshuk, & Patel, A. (2003). *Cognitive trait model-a supplement to performance based student models.* Paper presented at the International Conference on Computers in Educatio, Hong Kong, China.

Litzinger, T. A., Lee, S. H., & Wise, J. C. (2005). A study of the reliability and validity of the Felder-Soloman Index of Learning Styles. *Education, 113*, 77.

Livesay, G., Dee, K., Nauman, E., & Hites Jr, L. (2002). *Engineering student learning styles: a statistical analysis using Felder's Index of Learning Styles.* Paper presented at the ASEE Conference and Exposition, Montreal, Quebec.

Lopes, W. M. G. (2002). *ILS-inventário de estilos de aprendizagem de Felder-Saloman: investigação de sua validade em estudantes universitários de Belo Horizonte.* Universidade Federal de Santa Caterina, Brazil.

Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y., & Tan, C. L. (2003). Ant colony optimization for design of water distribution systems. *Journal of water resources planning and management, 129*(3), 200-209.

Mampadi, F., Chen, S. Y., Ghinea, G., & Chen, M.-P. (2011). Design of adaptive hypermedia learning systems: A cognitive style approach. *Computers & Education, 56*(4), 1003-1011.

Mayer, R. E., & Moreno, R. (1998). A split-attention effect in multimedia learning: Evidence for dual processing systems in working memory. *Journal of Educational Psychology, 90*(2), 312.

Mayer, R. E., & Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist, 38*(1), 43-52.

McArthur, D., Stasz, C., Hotta, J., Peter, O., & Burdorf, C. (1988). Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional science, 17*(4), 281-307.

McCalla, G. I., Bunt, R. B., & Harms, J. J. (1986). The design of the SCENT automated advisor. *Computational Intelligence, 2*(1), 76-92.

McVay, J. C., & Kane, M. J. (2012). Why does working memory capacity predict variation in reading comprehension? On the influence of mind wandering and executive attention. *Journal of Experimental Psychology: General, 141*(2), 302.

Melis, E., Andres, E., Budenbender, J., Frischauf, A., Goduadze, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2001). ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education (IJAIED), 12*, 385-407.

Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review, 63*(2), 81.

Mitchell, M. (1998). *An introduction to genetic algorithms*: MIT press.

Mitchell, T. (1997a). Artificial neural networks *Machine Learning* (Vol. 45, pp. 81-127). Burr Ridge, IL: McGraw Hill.

Mitchell, T. (1997b). *Machine Learning* (Vol. 45). Burr Ridge, IL: McGraw Hill.

Moreno, R. (2004). Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science, 32*(1-2), 99-113.

Mulwa, C., Lawless, S., Sharp, M., Arnedillo-Sanchez, I., & Wade, V. (2010). *Adaptive educational hypermedia systems in technology enhanced learning: a literature review.* Paper presented at the Proceedings of the 2010 ACM conference on Information technology education.

Myers-Briggs, I. (1962). *The Myers-Briggs type indicator: manual*. Palo Alto, CA: Consulting Psychologists Press.

Nakamichi, Y., & Arita, T. (2004). Diversity control in ant colony optimization. *Artificial Life and Robotics, 7*(4), 198-204.

Niles, L., Silverman, L. N. H., Tajchman, G., & Bush, M. (1989). *How limited training data can allow a neural network to outperform an optimal statistical classifier.* Paper presented at the Internationall Conference on Acoustics, Speech, and Signal Processing.

Özpolat, E., & Akar, G. B. (2009). Automatic detection of learning styles for an e-learning system. *Computers & Education, 53*(2), 355-367.

Paas, F. G. (1992). Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology, 84*(4), 429.

Paas, F. G., Renkl, A., & Sweller, J. (2004). Cognitive load theory: Instructional implications of the interaction between information structures and cognitive architecture. *Instructional science, 32*(1), 1-8.

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2003). Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. *User Modeling and User-Adapted Interaction, 13*(3), 213-267.

Pashler, H., McDaniel, M., Rohrer, D., & Bjork, R. (2008). Learning styles concepts and evidence. *Psychological science in the public interest, 9*(3), 105-119.

Pask, G. (1976). Styles and strategies of learning. *British Journal of Educational Psychology, 46*(2), 128-148.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*(1), 33.

Pollack, I. (1953). The information of elementary auditory displays. II. *The Journal of the Acoustical Society of America, 25*(4), 765-769.

Popescu, E. (2010). Adaptation provisioning with respect to learning styles in a Web-based educational system: an experimental study. *Journal of Computer Assisted Learning, 26*(4), 243-257.

Pothiya, S., Ngamroo, I., & Kongprawechnon, W. (2010). Ant colony optimisation for economic dispatch problem with non-smooth cost functions. *International Journal of Electrical Power & Energy Systems, 32*(5), 478-487.

Priddy, K. L., & Keller, P. E. (2005). *Artificial neural networks: an introduction* (Vol. 68): SPIE Press.

Rajendran, C., & Ziegler, H. (2004). Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European journal of operational research, 155*(2), 426-438.

Riget, J., & Vesterstrøm, J. S. (2002). A diversity-guided particle swarm optimizer-the ARPSO. *Dept. Comput. Sci., Univ. of Aarhus, Aarhus, Denmark, Tech. Rep, 2*, 2002.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: a modern approach* (Vol. 3). Upper Saddle River: Prentice Hall.

Schmeichel, B. J., Volokhov, R. N., & Demaree, H. A. (2008). Working memory capacity and the self-regulation of emotional expression and experience. *Journal of personality and social psychology, 95*(6), 1526.

Seery, N., Gaughran, W., & Waldmann, T. (2003). *Multi-modal learning in engineering education.* Paper presented at the 2003 ASEE Conference and Exposition.

Segedy, J. R., Biswas, G., Blackstock, E. F., & Jenkins, A. (2013). *Guided skill practice as an adaptive scaffolding strategy in open-ended learning environments.* Paper presented at the Artificial Intelligence in Education.

Self, J. A. (1994). Formal approaches to student modelling *Student modelling: The key to individualized knowledge-based instruction* (pp. 295-352): Springer.

Shi, Y., & Eberhart, R. C. (1998, March 25-27). *Parameter selection in particle swarm optimization.* Paper presented at the Evolutionary Programming VII, San Diego, CA.

Shmygelska, A., & Hoos, H. H. (2005). An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC bioinformatics, 6*(1), 1.

Sison, R., & Shimura, M. (1998). Student modeling and machine learning. *International Journal of Artificial Intelligence in Education (IJAIED), 9*, 128-158.

Spurlin, J. (2002). Unpublished data.

Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer, 27*(6), 17-26.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation, 10*(2), 99-127.

Sun, R., & Peterson, T. (1998). Autonomous learning of sequential tasks: experiments and analyses. *Neural Networks, IEEE Transactions on, 9*(6), 1217-1234.

Surjono, H. D., & Maltby, J. R. (2003). Adaptive educational hypermedia based on multiple student characteristics *Advances in Web-Based Learning-ICWL 2003* (pp. 442-449): Springer.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive science, 12*(2), 257-285.

Swingler, K. (1996). *Applying neural networks: A practical guide*. San Franciso, CA: Morgan Kaufmann.

Teigen, K. H. (1994). Yerkes-Dodson: A law for all seasons. *Theory & Psychology, 4*(4), 525-547.

Thomas, L. F., & Harri-Augstein, E. S. (1977). Learning to learn: the personal construction and exchange of meaning. *Adult learning*.

Tuckman, B. W., & Harper, B. E. (2012). *Conducting educational research*: Rowman & Littlefield Publishers.

Tuholski, S. W., Engle, R. W., & Baylis, G. C. (2001). Individual differences in working memory capacity and enumeration. *Memory & Cognition, 29*(3), 484-492.

Turner, M. L., & Engle, R. W. (1989). Is working memory capacity task dependent? *Journal of memory and language, 28*(2), 127-154.

Ueno, M. (2005). *Intelligent LMS with an agent that learns from log data.* Paper presented at the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.

Unsworth, N., Redick, T. S., Spillers, G. J., & Brewer, G. A. (2012). Variation in working memory capacity and cognitive control: Goal maintenance and microadjustments of control. *The Quarterly Journal of Experimental Psychology, 65*(2), 326-355.

Van Merriënboer, J., Schuurman, J., De Croock, M., & Paas, F. (2002). Redirecting learners' attention during training: Effects on cognitive load, transfer test performance and training efficiency. *Learning and instruction, 12*(1), 11-37.

Van Zwanenberg, N., Wilkinson, L. J., & Anderson, A. (2000). Felder and Silverman's Index of Learning Styles and Honey and Mumford's Learning Styles Questionnaire: How do they compare and do they predict academic performance? *Educational Psychology, 20*(3), 365-380.

VanLehn, K. (1988). Student modeling. *Foundations of intelligent tutoring systems, 55*, 78.

Villaverde, J. E., Godoy, D., & Amandi, A. (2006). Learning styles' recognition in e-learning environments with feed-forward neural networks. *Journal of Computer Assisted Learning, 22*(3), 197-206.

Walkington, C. A. (2013). Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology, 105*(4), 932.

Wanas, N., Auda, G., S. Kamel, M., & Karray, F. (1998, May). *On the optimal number of hidden nodes in a neural network.* Paper presented at the IEEE Canadian Conference on Electrical and Computer Engineering.

Wermter, S., & Sun, R. (2000). An Overview of Hybrid Neural Systems *Hybrid Neural Systems* (pp. 1-18). New York: Springer Science & Business Media.

Willcoxson, L., & Prosser, M. (1996). Kolb's Learning Style Inventory (1985): Review and further study of validity and reliability. *British Journal of Educational Psychology, 66*(2), 247-257.

Wilson, D. (1986). An investigation of the properties of Kolb's Learning Style Inventory. *Leadership & Organization Development Journal, 7*(3), 3-15.

Woehrle, J. L., & Magliano, J. P. (2012). Time flies faster if a person has a high working-memory capacity. *Acta psychologica, 139*(2), 314-319.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE, 87*(9), 1423-1447.

Yao, X., & Liu, Y. (1997). A new evolutionary system for evolving artificial neural networks. *Neural Networks, IEEE Transactions on, 8*(3), 694-713.

Yu, P., Own, C., & Lin, L. (2001). *On learning behavior analysis of web based interactive environment.* Paper presented at the International Conference on Implementing Curricular Change in Engineering Education.

Zigmond, M. J., & Bloom, F. E. (1999). *Fundamental neuroscience*. San Diego, CA, USA: Academic Press.

Zywno, M. S. (2003). *A contribution to validation of score meaning for Felder-Soloman's index of learning styles.* Paper presented at the American Society for Engineering Education Annual Conference & Exposition.