

Programming by Example; A technique to automate database query

Mahin Talukder ⁹

Abstract

The paper “Programming by Example; A technique to automate database query” encourages non-programmers to develop programming skills and natural language processing. The traditional approach of manually constructing a programme for such a normalisation task requires expertise in both programming and target (human) language and further does not scale to a large number of domain, format or target language combinations. This article proposes programmes can be learned for such normalisation tasks through examples. A domain specific programming language offers appropriate abstractions for succinctly describing such normalisation tasks, and then present a novel search algorithm that can effectively learn programs in this language from input-output examples. In this paper we describe domain-specific heuristics for guiding researchers to normalisation tasks related to that domain.

Keywords: Programme Synthesis, Programming by Examples, Inductive Synthesis, Deductive Synthesis, MySQL Programming, End-user Programming

Introduction

Nowadays there are several strategies to store data in computers. Data stored in personal computers is different from storing data in the Cloud¹⁰ or server computer. As an end user, data keeper, we have interest in our information only but behind the screen we are not concerned about the process of data storage or retrieving information. In order to store large scale data, a relational database management system (RDBMS) is used to maintain data consistency, followed by structure. A computer expert might have knowledge about how to write queries in a computer language for data storage and extract information out of a large data scale but general people lack this knowledge. As a result, it gets harder for users to produce a list of necessary information from a given data set.

Programming by example is a technique to help those people who do not have sufficient knowledge about programming. A system would consider user intention from user, by examples. Considering those example the system could perform an automated search throughout the database. Considering the example of a given statement, the system could generate probabilistic interest of user search. By using such system, the user can enhance search operation without having sufficient knowledge about computer programme or programming language.

During a work placement at a leading private college and being a part of the IT and monitoring team, the author has participated in every aspect of development and report management. Using a database management system (DBMS), with its graphical user interface, requires database query to generate information. Sometimes it is more complicated while using a relational database management system (RDBMS). As a computer science graduate, it has been easier for me to write the programming code (e.g. SELECT or UPDATE statements) than for those people lacking programming knowledge or skills.

Humans are good with examples. They can express examples to determine the nature of their interest. A system that takes user interest from examples and creates a possible combination for those examples to complete the required list, would illuminate this. Transforming user thoughts (processing a natural language) into a database query can make the information available for everybody, specifically people with less skills of programming languages.

⁹ PhD Candidate

¹⁰ Cloud is synonymous with Apple iCloud, which is a popularly used Internet based data storage system for individual and company use

Motivation for this paper

'Flash Fill' is a feature introduced by Microsoft [MS] Excel as part of their package MS Office 2013. In this feature, MS Excel's columns consider a pattern while extracting portion of a data. Applying the few examples of data extraction or combination, the 'Flash Fill' feature follows the same pattern for all remaining data.

While we consider the process precisely we can find that for every element that 'Flash Fill' extracts or inserts requires a unique programming mechanism. With the help of 'Flash Fill', MS Excel completes the programming for human understanding by considering the few examples provided to MS Excel. This paper uses the same strategy to identify user required information from selective databases.

During bachelor degree studies, the author has undertaken a project on a student database management system. The process required adding several students and staffs to a database. Student information was organised according to semester, course and modules. Students were assigned to classes by status which made them available in the attendance list. The result list was constructed, taking into consideration students' assignment submissions and examination results. A single student could appear in different lists according to selected criteria. Staff were provided access to these students in accordance with their permission level. The project was completed using PHP¹¹ and MySQL¹², which may be considered as computer languages.

During my master degree, the author has discovered more modern approaches of computation. Modules taken have expand my knowledge on artificial intelligence, natural computation and cloud computing. Advanced Java¹³, C++¹⁴ and logic¹⁵ programming has developed my skills in programming languages. The MSc project I undertook was 'Query

Expansion with Web Search' which has given me an opportunity to enhance search operations within local and global environments. Study on query modelling, query expansion and various information retrieval technique has lead the author to investigate this paper's concept.

Information can be considered a stimulus that has meaning in some context for its receiver. Information entered and stored electronically is considered data (Withers, 2005). When data is processed it is perceived as information (Nicolaou *et al.*, 2006). Nowadays there are various strategies to store data and extract information out of the data. Use of database is most commonly used to store data and produce information. Programmers have to write interactive code to produce required information. Those people with less programming skills find it difficult to write specific pieces of code to generate information. There are many different types of DBMSs, ranging from small systems that run on personal computers to vast systems that run on mainframes. Sometimes it can become extremely complicated to write programmes to extract information from different types of DBMS (Kepler *et al.*, 2004).

In computer science, the term 'automatic programming' (Mur *et al.*, 2006) identifies a type of computer programming in which some mechanisms generate a computer program to allow human programmers to write codes at higher abstraction levels. Over time the true meaning of 'automatic programming' has changed and nowadays any programme that uses Artificial Intelligence (AI) to compute programmes is considered as 'automatic programming' (Perry and Douglas, 2002).

Substantial research is taking place to automate programming language. 'Automatic programming' is based on specifications that are higher-level and easier for humans to specify than ordinary programming languages (Weiss and Gerhard, 1999). Few researchers

¹¹ PHP is a server-side scripting language designed primarily for web development.

¹² MySQL is an open-source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).

¹³ Java is a concurrent, class-based, object-oriented computer programming language

¹⁴ C++ is a programming language standardized by the International Organisation for Standardization (ISO)

¹⁵ Logic programming is a type of programming paradigm which is largely based on formal logic

have combined Artificial Intelligence with a compiler to automate the process. It has been summarised in the research paper “*Artificial Intelligence meets Compilers*” where human or

machine write programme (execute) based on knowledge of algorithms, data structures, design patterns etc. (Browne *et al.*, 1989; Weiss and Gerhard, 1999).

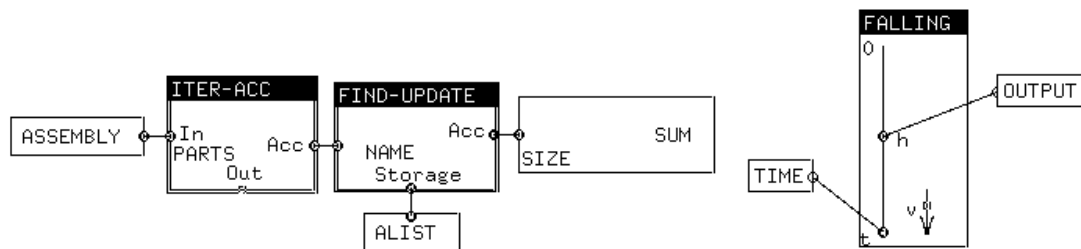


Fig 1: Generalising compile process (Browne *et al.*, 1989)

Programming by Example projects would use AI mechanism to accelerate computing by creating an automatic database query. AI techniques should be initiated to represent, find, and substantiate the design patterns to automate the programme. A programme that requires multiple components needs to combine these to accomplish the desired task.

The belief that ‘...programming hasn’t changed in 30 years...’ (Browne *et al.*, 1989) can be challenged. Power-based Strategy for AI uses speed and power of the computer to derive answers to problems by search, starting from a small number of principles. On the other hand knowledge-based Strategy solves problems which is derived from large amounts of knowledge (Xing *et al.*, 2010). The computer must similarly embody knowledge to be a good problem-solver. This strategy is used to express knowledge or to solve new problem based on experience.

‘Automatic programming’ approach is the evolutionary algorithm engine that was first described by John Holland in 1975 and inspired by the notions of survival of fittest (Spencer, 1864) and Darwinian natural selection (O’Neill and Ryan, 2012). A population of candidate-solution is maintained, each individual solution is evaluated to determine its performance on problem-solving. This approach is mostly known as ‘search algorithm’. Most of the evolutionary algorithms are designed to solve a particular problem, evolve by themselves, but

however in some case the search operation becomes inhibited in local optimisation, when a better solution might exists globally.

Automatic string processing is a vast area of research. String data is ubiquitous and data management has become particularly significant in the past ten years. Approximate queries are very important on string data due to typographical errors in data (Gravano *et al.*, 2001). Commercial databases do not support approximate string queries directly. Thus an effort has been made to implement approximate string processing technique with user-defined functions (UDFs). The process is called ‘q-grams’ which was implemented by exploiting facilities already available in database (Gravano *et al.*, 2001). However the technique of ‘q-grams’ does not take user input as an example from data collection and produce information as required by users.

Trillium Software is a commercial company¹⁶ (Friedman and Bitterer, 2005) which integrates databases for different companies. Trillium Software systems provide services to maintain scalability and data consistency between different databases within an organisation. As a result user data maintain consistency throughout different systems. However, computer scientists have provided several algorithm considering matches on specific types of data but those mechanisms have not been integrated with commercial database (Navarro *et al.*, 2011). However, an approach

¹⁶ [weblink: <http://www.trilliumsoftware.com>]

has been proposed to join approximate strings within a database. 'Approximate equality' is very popular and describes the notion of edit distance between two strings. According to this notion, deletion, insertion and substitution of a character are considered as unit cost operations and the edit distance between two strings is defined as the lowest cost sequence of operations that can transform one string to the other (Gravano, *et al.*, 2001). This mechanism could be beneficial when considering user data as an example for database query.

Over the last several years, demand for integrating full text and relational search has increased dramatically. IBM's¹⁷ Digital Library exploited DB2¹⁸ and IBM's powerful linguistic text search engine [TSE] decomposed queries into relational and text search components in application layers (Maier and Simmen, 2001). DB2's Text Extender (TE) was the first attempt by a major RDBMS vendor to tackle search problems. TE exploited DB2's SQL¹⁹ extensibility features to effectively push query decomposition and join into RDBMS. The table function text search calls TSE to evaluate the text search condition to determine the search relevance. This approach allows text and relational results to be joined using the traditional RDBMS table joining method (Maier and Simmen, 2001).

Microsoft SQL Server and Oracle Text Retrieval have introduced a function that takes parameters as a key and returns an 'inverted list' index (Hamilton and Nayak, 2001; Dixon, 2001). This is a regular scan of an expression which works as a framework in SQL language. Desired information is required to be independent of similarity, subsequence, typing or spelling mistake. There are several proposal that have been presented to index text searches to speed up approximate searches. These include 'Suffix tree', 'Suffix Array', 'Q-grams' and 'Q-samples' models which contain different data structures. Search approaches within these structures are different considering index terms, number of appearances and considering neighbourhood generation

(Gravano, 2001). More research is required to establish how competitive these ideas could be in programming by example project.

In 1945 Vannevar published a ground breaking article titled 'As We May Think' which gave birth to the idea of automatic access to large amounts of stored knowledge (Vannevar, 1945) Management Information Systems (MIS) are classified based on the primary design to handle data. Relational or object-oriented databases are designed for strictly typed data and semi-structured database are designed to manage data efficiently that partially conforms to a schema (Hamilton and Nayak, 2001; Dixon, 2001). Minimal programming knowledge is required to extract relevant information from these data structures.

MS Excel is a platform developed by Microsoft to enhance various processes, including data-sorting, calculation, graphing tools and pivot tables. MS Excel uses computer programmes to manipulate and organise data. Spreadsheets in MS Excel present table of values arranged in rows and columns which is quite similar to modern databases. Values of row and column can be manipulated mathematically using arithmetic operations and functions. Moreover, macro programming language can be used to manipulate data in rows and columns in the spreadsheet. Writing scripts or macros can be difficult for a person who does not have any skills or knowledge of Visual Basic or .Net programming language. Considering this kind of scenario, Microsoft has developed an inductive synthesis, methodology that has enabled synthesisers to generate a wide range of string/table manipulating programs in spreadsheets from input-output examples automatically (Marcotte, 1999). The four major methodologies used to determine output from examples are '*...Domain-specific language, Data structure for representing consistent programs, Algorithm for synthesising consistent programs and Ranking...*' (Gulwani, *et al.*, 2012). The system considers real-world task by taking examples from the user and finds consistency with a given data set. An algorithm for synthesising consistent programmes is used

¹⁷ IBM is an American multinational technology company that produce computer component and business solutions

¹⁸ DB2 is a database product from IBM that uses Relational Database Management System (RDBMS)

¹⁹ SQL: Structured Query Language

to generate or intersect data with existing examples and the process continues until the example data remains as a set of data structure for representing consistent programs. Finally a data set is ranked according to the examples. These methodologies can be used for modern databases. Behind the scenario, the system would generate a list of automated query that finds possible combination from given examples.

Programming by example is a type of project that automates user interest of search, considering examples to the system. The system would be integrated with traditional databases to extract information from the database. A successful completion of the project would automate SQL queries within the system. Automating the program is a vast area of research. Hence there are number of companies who have adopted this kind of strategy. These include:

- Kestrel Institute / Reasoning Systems, Cordell Green and Douglas Smith have introduced generations of programmes from formal specifications (predicate calculus, category theory). This has driven success for combinatorics search problems (Smith *et al.*, 1985).
- SciComp (Austin), Elaine Kant has used generation of code to simulate systems of differential equations. It was originally seismic analysis, now evaluation of derivative securities (Young *et al.*, 2000).
- Semantic Designs (Austin), Ira Baxter has used advanced program manipulation to improve commercial software (Kuipers and Byun, 1991).
- Intentional Software Corp., Charles Simonyi has produced advance software tools for programmers' use (Simonyi *et al.*, 2006).
- NASA Ames, In Automated Software Engineering, Mike Lowry has used predicate calculus representation of scientific subroutines to find ways to use the subroutines to accomplish a goal in spacecraft navigation and control (Hart and Staveland, 1988).

Current Statue of Art in Programming by example

A traditional view of program synthesis is that of synthesis from complete specifications. One approach is to give a specification as a formula in a suitable logic and another is to write the specification as a simpler, but possibly far less efficient program. While these approaches have the advantage of completeness of specific-cation, such specifications are often unavailable, difficult to write, or expensive to check against using automated verify-cation techniques. Jha *et al.* (2010) proposed a novel oracle guided approach to program synthesis, where an I/O oracle that maps a given program input to the desired output is used as an alternative to having a complete specification. The key idea of our algorithm is to query the I/O oracle on an input that can distinguish between non-equivalent programmes that are consistent with the past interaction with the I/O oracle. The process is repeated until a semantically unique program is obtained.

Various language-specific techniques have also been proposed for French (Larreur *et al.*, 1989), Russian (Sproat, 2010) and Croatian (Beliga and Martincic-Ipsic, 2011). Machine translation based methods have also been attempted (Schlippe *et al.*, 2010) but these and other statistical translation methods require thousands of examples before they converge on approximately correct solutions. The upside to statistical methods is the ability to capture noisy input data that our methods do not possess. But 'Flash Fill' is much more useful when the user wants to provide a relatively small sample and can guarantee that it is consistent.

The most accepted technical point of transition proposed by Sumit (2011) is focused on synthesising spreadsheet programs. It treats text normalisation as a string manipulation task, but a more sophisticated one than what has been attempted in 'Flash Fill'. The task requires (a) learning larger programmes in a more expressive DSL that is extensible and allows table lookups and (b) dealing with more elaborate specification in the form of large number of examples (Kini and Gulwani, (2015)

Research Objectives

Any systematic investigation into and study of materials and sources in order to establish facts and reach new conclusions, is considered as research. Despite all the research literature reviewed there are the following areas that need special attention to complete a 'Programming by Example' project. Research objectives will be determined considering the various potential scopes of the project.

Relational Database Management System (RDBMS) is one of modern Database Management System (DBMS) used by many popular databases. In RDBMS database, the structure is based on a relational model invented by Codd and developed in IBM's research (Carrasco et al., 2003). For the purpose of this project research needs to be carried out to understand the structure and mechanism to retrieve information, mostly query various database statements (including SELECT, UPDATE, DELETE and INSERT).

Companies of all sizes including non-commercial users need sophisticated data management programmes. The nature of the business and number of IT professional within their team would determine how efficient the data management project would be. It will also determine requirements specification of the system. In the preliminary phase of the project, primary research techniques would be adopted using questioner and survey. Social media, email, telephone call as a medium of communication, might be used during this process.

Finally, most importantly, research would be carried out on existing models, algorithms and approaches to process example query. For this project, our focus is on finding key information using natural language (examples from database) and relating key words to our database. The results would be produced based on anonymous patterns that exist in all examples. Eventually, an increased number of examples would generate precise and specific information.

During this project, precise focus is given to the following criteria and objectives:

A mechanism to upload SQL (can be a web based application or desktop based application or a plugin that can be added to any website.)

- Extract SQL into the system
- Read the table and column name from SQL
- String processing and handling
- Identify relevant properties for Examples
- Present produced information in an orderly manner.

Research Questions

The biggest question asked by researcher in this field is 'Can we make programme automated?' or 'Can any system really identify user interest of search'. Answers to this type of question will be revealed by a 'Programming by Example' project. In addition, the following questions will represent an evaluation of the computing industry.

- What are the main challenges to process examples into a programming language?
- What kinds of database searches are people undertaking?
- Why do we need to produce information for our own interest?
- Is data consistency really important?
- Are we able to minimise high level programming?
- How will organisation benefit by such systems?
- Does this kind of system accelerate business decision?
- What is preventing users to undertake a search for their interest?
- How programme adaptability responds to people with less programming knowledge or skills?

Research Methodology

In the methodology I use a number of approaches to delve further into the project. Primary research would be on identifying types of search operations that are performed within a database for small, medium and large organisation. Data will be collected from different stakeholders by interviewing personally, by phone or email. In this case judgement sampling will be used to identify obstacles to the search.

Open source, existing algorithms and approaches on machine learning, neural networks, evolutionary algorithms and genetic

programming will be considered as secondary research for the 'Programming by Example' project.

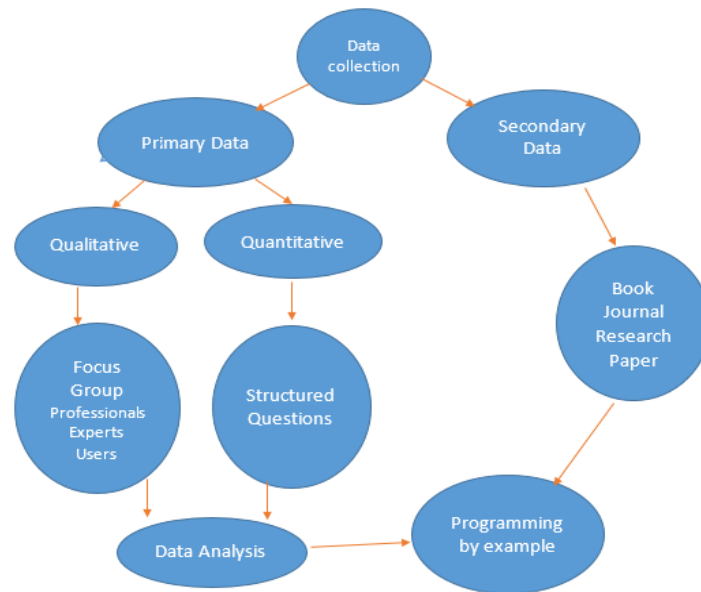


Fig. 2: Research strategy for 'Programming by example'

The key methodologies for this project that could be used are "Experimental Methodologies" and 'Build Methodology'. Experimental Methodologies would help to keep a track of experimental setup and record keeping. Record keeping tends to be surprisingly lax and can be used concurrently by different researcher. Experimental setup and results could help to identify mislead or distort data. On the other hand Build Methodology requires system design, system architecture and a reused component from a selective programming language (for example 'Java'). This approach would keep examples as input and sorted SQL result list as output around the system to perform unit testing. Concurrent Version System (CVS) would provide suitable accessibility to continue the project for an extended time. Thus it opens a new window for further development.

Conclusion

Artificial intelligence and DBMS are two different, well-established technologies. Integration of artificial intelligence in DMBS that would play a significant role for future computing. By establishing a co-relation between AI and DBMS, a revolution will be marked for the next generation of computing. Examples are often a natural way to specify various computational artefacts such as programs, queries and sequences. Synthesising such artefacts from example-based specifications has various applications in the domains of end-user programming and intelligent tutoring systems. Synthesis from examples involves addressing two key technical challenges: (i) design of a user interaction model to deal with the inherent ambiguity in the example based specification and (ii) design of an efficient search algorithm – these algorithms have been based on

paradigms from various communities. The process will enable the user (a) to share larger amount of data for knowledge processing, (b) efficient management of data as well as knowledge and (c) intelligent data processing. The Programming by Example

Project is a small area of information processing with artificial intelligence which aims to store, analyse and retrieve the data efficiently. Much further research needs to be undertaken to develop this work into a useable tool for user programming.

References

- Browne, J.C., Asam, M. and Sobek, S., (1989). Code: A unified approach to parallel programming. *Software, IEEE*, **6**(4), pp. 10-18.
- Bush, V. (1945), 'As we may think', *Atlantic Monthly* (7), 101-108.
- Carrasco, R.A., Vila, M.A. and Galindo, J., (2003). FSQL: a flexible query language for data mining, *Enterprise information systems IV 2003*, Kluwer Academic Publishers, pp. 68-74.
- Dixon, P., (2001). Basics of oracle text retrieval. *IEEE Data Eng.Bull.*, **24**(4), pp. 11-14.
- Friedman, T. and Bitterer, A., (2005). Market Scope Update: Data Quality Technology.
- Gravano, L., Ipeirotis, P.G., Jagadish, H.V., Koudas, N., Muthukrishnan, S., Pietarinen, L. and Srivastava, D., (2001). Using q-grams in a DBMS for approximate string processing. *IEEE Data Eng.Bull.*, **24**(4), pp. 28-34.
- Gulwani, S., Harris, W.R. and Singh, R., (2012). Spreadsheet data manipulation using examples. *Communications of the ACM*, **55**(8), pp. 97-105.
- Hamilton, J.R. and Nayak, T.K., (2001). Microsoft SQL server full-text search. *IEEE Data Eng.Bull.*, **24**(4), pp. 7-10.
- Hart, S.G. and Staveland, L.E., (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, **52**, pp. 139-183.
- Jha, S.; Gulwani, S.; Seshia, S. A. & Tiwari, A. (2010). Oracle-guided component-based program synthesis., in Jeff Kramer; Judith Bishop; Premkumar T. Devanbu & Sebastián Uchitel, eds., 'ICSE (1)', ACM, pp. 215-224 .
- Kepler, M.A., Huey, C.A., Qi, R. and Wake, C.A., (2004). Method and system for retrieving data from multiple data sources using a search routing database, USA: Grant, <http://www.google.com/patents/US6745177> accessed 01 January 2016.
- Kini, D. & Gulwani, S. (2015). FlashNormalise: Programming by Examples for Text Normalisation in Qiang Yang & Michael Wooldridge (eds.), 'IJCAI', AAAI Press, pp. 776-783
- Kuipers, B. and Byun, Y., (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, **8**(1), pp. 47-63.
- Maier, A. and Simmen, D.E., (2001). DB2 optimisation in support of full text search. *IEEE Data Eng.Bull.*, **24**(4), pp. 3-6.
- Marcotte, E.M., Pellegrini, M., Thompson, M.J., Yeates, T.O. and Eisenberg, D., (1999). A combined algorithm for genome-wide prediction of protein function. *Nature*, **402**(6757), pp. 83-86.
- Mur, R.A., (2006). *Automatic Inductive Programming*, Proceedings of the 23rd international conference on machine learning, tutorial 2006.
- Navarro, G., Baesa-Yates, R.A., Sutinen, E. and Tarhio, J., (2001). Indexing methods for approximate string matching. *IEEE Data Eng.Bull.*, **24**(4), pp. 19-27.
- Nicolaou, A.I. and McKnight, D.H., (2006). Perceived information quality in data exchanges: Effects on risk, trust, and intention to use. *Information Systems Research*, **17**(4), pp. 332-351.
- O'Neill, M. and Ryan, C., (2012). Grammatical evolution: evolutionary automatic programming in an arbitrary language. London: Springer Science & Business Media.
- Perry, D.L., 2002. *VHDL: Programming by Example*, New York: McGraw-Hill.
- Simonyi, C., Christerson, M. and Clifford, S., (2006). Intentional software, *ACM SIGPLAN Notices* 2006, ACM, pp. 451-464.
- Smith, D.R., Kotik, G.B. and Westfold, S.J., (1985). Research on knowledge-based software environments at Kestrel Institute. *Software Engineering, IEEE Transactions on*, (11), pp. 1278-1295.

Weiss, G., (1999). Multiagent systems: a modern approach to distributed artificial intelligence. USA: MIT Press.

Withers, K.J., (2005). Electronically stored information: The December 2006 amendments to the federal rules of civil procedure. *New Journal of Technology & Intelligence. Prop.*, 4, pp. 171.

Xing, L., Chen, Y., Wang, P., Shao, Q. and Xiong, J., (2010). A knowledge-based ant colony optimisation for flexible job shop scheduling problems. *Applied Soft Computing*, 10(3), pp. 888-896.

Young, R.L., Kant, E. and Akers, L.A., (2000). A knowledge-based electronic information and documentation system, Proceedings of the 5th international conference on intelligent user interfaces, ACM, pp. 280-285.