**Genetic Algorithm For Timetable Generation**

**Abhay Shanker Pathak**

(Student)

Department of

Information Technology

United Institute of Technology

**Pragati Srivastava**

(Student)

Department of

Computer Science and Engineering

United Institute of Technology

**Tarun Singh**

(Student)

Department of

Computer Science and Engineering

United Institute of Technology

**Dr. Umesh Kumar Pandey**

(Assistant Professor)

Department of

Computer Science and Engineering

United Institute of Technology

- - - - - - - - - - - - - - υ - - - - - - - - - - - - - -

**ABSTRACT**

Time-table is created for various purposes like to organize for lectures in school and colleges, to create timing charts for train and bus schedule and many more.To create time-table manually it takes lots of time and man power. For finding a feasible time-table for heavy constrains is very challenging and error prone. In this paper Tigen has been made to resolve these difficulties which is based on the concept of Genetic Algorithm (Evolutionary Algorithm). Genetic Algorithm is a part of Artificial Intelligence. Our project uses this as a core. Being an NP-complete problem, many attempts have been made using varying computational methods to obtain optimal solutions to the timetabling problem. Genetic algorithms, based on Darwin's theory of evolution is one such method. The aim of this study is to optimize a general university course scheduling process based on genetic algorithms using some defined constraints. The approach uses a problem specific chromosome representation. heuristic approach is used for obtaining feasible solution. An intelligent mutation scheme has been employed for speeding up the convergence. Apart from this, the project uses some other traditional and non-traditional methods like interaction with database system and providing custom data-structure for storage of data during program execution etc. for successful interaction and getting result from project.The comprehensive tigen presented in this paper has been validated tested and discussed using real world data from our college.

**INTRODUCTION**

Tigen is basically the scheduling and assignment of the entities into appropriate time slots and resource respectively without causing time clashes for the students and the teachers ,as well as the resource clashes same as in other areas. This task is very slow and laborious performed by people working on the strength of their knowledge of resources and constraints of a specific institution. Tigen is the schedule (time-table) generator which is based on the concept of Genetic Algorithm (Evolutionary Algorithm) which is a part of Artificial Intelligence.

Genetic Algorithm is used to get solutions for problem inspired by nature. Genetic Algorithm (GA) is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems, and find feasible solutions

Feasible solutions here mean those which do not violate hard constraints and as well try to satisfy soft constraints. Hard constraints concern issues that are physically impossible – such as a teacher or student being in two places at the same time. Similarly, two teachers are not permitted to 2 teach two separate courses to the same group of students in a given time slot. Further, allocations of two or more classrooms are not made for the same course for a given group of students. Thus, it is necessary that, neither the staff nor the students can be in more than one place at a given time. Further, all the necessary resources such as the staff, rooms etc. should be available for each time slot. We need to choose the most appropriate one from feasible solutions. Most appropriate ones here mean those which do not violate soft constraints to a greater extent. soft constraints include the assignment of “hard courses” in time slots in the morning sessions when students are able to pay attention to such subjects. When higher number of students occupy large classrooms, changing rooms after every lecture is avoided unless small groups have to move to small spaces such as laboratories. Staff preferences such as teaching at times and classrooms of choice are additional constraints. Using Genetics Algorithm, a number of trade-off solutions, in terms of multiple objectives of the problem, could be obtained very easily. Moreover, each of the obtained solutions has been found much better than a manually prepared solution which is in currently in use. Although manual scheduling is time consuming and inaccurate, small universities adapt to generate their schedules. As the complexity of university increases it to become necessary to adopt computer methods to ease the task of timetabling.

**REVIEW OF LITERATURE**

There are many other schedulers present on the internet that do a job similar to that of our software. But they are not what our project is intended to be. The solutions below mentioned are mostly used as purpose of demonstrating a use case for Genetic (Evolutionary) Algorithm.

According to us, best solutions from the below mentioned existing solutions is by pranavkhurana, but it is still far from what our project is ought to be.

Some of the existing solutions on Github are:

* nuhu-ibrahim/time-table-scheduling (<https://github.com/nuhu-ibrahim/time-table-scheduling.git>)
* akazuko/timetablescheduler (https://github.com/akazuko/timetablescheduler.git)
* Baksonator/evolutionary-timetable-scheduling (Baksonator/evolutionary-timetable-scheduling.git)
* pranavkhurana/Time-table-schedular (https://github.com/pranavkhurana/Time-table-schedular.git

**REQUIREMENT ANALYSIS**

Software requirement is a functional or non-functional need to be implemented in the system. Functional means providing particular service to the user. For example, in context to banking application the functional requirement will be when customer selects “View Balance” they must be able to look at their latest account balance.

Software requirement can also be a non-functional, it can be a performance requirement. For example, a non-functional requirement is where every page of the system should be visible to the users within 5 seconds. So, basically software requirement is a

• Functional

• Non-functional

need that has to be implemented into the system. Software requirement are usually expressed as a statements A focused and detailed requirements analysis can help you avoid problems like these. This is the process of discovering ,defining, and documenting the requirements that are related to a specific business objective. And it’s the process by which you clearly and precisely define the scope of the project, so that you can assess the timescales and resources needed to complete it.

**Functional Requirements**:

The project should be able to provide these following functional requirements:

* To maintain login database with correct login credentails
* To be able to show the schedule any time to end user .
* To perform user interaction successfully and take input values correctly.
* To provide a schedule from the provided data
* To show the schedule in correct format to end use.
* To insert, update or delete data to and from database successfully.

**Operational Requirements**:

The project should be able to perform these following operational requirements:

* To correctly output log and debug symbols to the file for debugging purpose
* To make a backup and preserve these log and debug files on the event of crash.
* To perform schedule generation algorithm process in case of update.
* To be able to synchronize the updated schedule with different users

**Technical Requirements:**

Following are the technical requirements which need to be fulfilled for the project:

* Availability of a "C/C++ compiler"
* Availability of a "MySQL DBMS system" in machine
* Availability of ncurses for Terminal user Interface
* Availability of QT for Graphical user Interface
* Availability of build tools like "CMake, Nmake, Unix-Makefiles"
* Availability of "git" if user wants the latest development support of project.

**PROPOSED APPROACH**

The Basic approach behind is to implement the complete flow of genetic algrorithm . In a genetic algorithm, a population of chromosomes consisting of a given random collection of genes is initiated according to the following steps.

* Generating an initial population of chromosomes.
* Evaluating the suitability of each chromosome (individual) that forms the population.
* Selecting the chromosomes for mating based on the above results.
* Producing offspring by mating (cross over) the selected chromosomes.
* Mutating genes randomly.
* Repeating steps 3-5 until a new population is generated.
* Ending the algorithm when the best solution obtained has not changed after a preset number of generations.

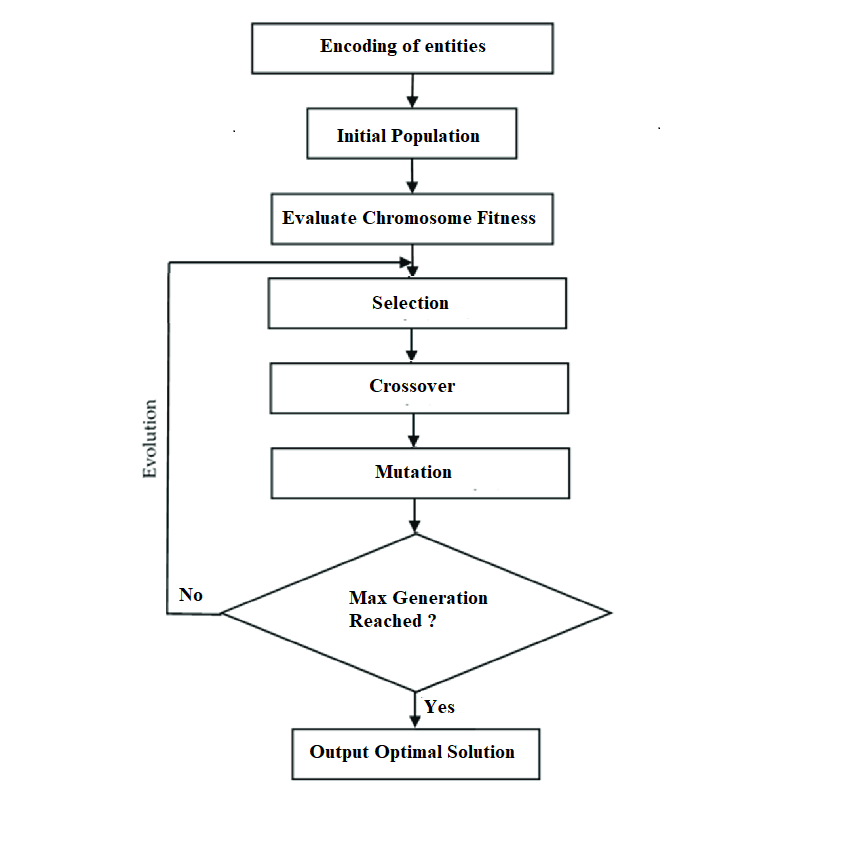
****

Figure 1: Genetic Algorithm Flowchart

**Encoding :**

As stated earlier, scheduling classes for a college timetable is often met with constraints (hard and soft) due to diversity as compared to a school timetable where the requirements are highly limited. The class-scheduling problem will be based on the available Instructors, classrooms, classtimes, Departments and courses. Each class will be assigned a timeslot, a professor, a room, and a course by the class scheduler. The total number of classes that needs to be scheduled can be obtained by summing the number of student groups multiplied by the number of modules each student group is enrolled in. For each class scheduled by this application the following hard constraints will be considered.

• Classes can only be scheduled in free rooms.

• A Instructor can only teach one class at any one time.

• Classrooms must be big enough to accommodate the Number of students in class.

When encoding the class schedule, certain class properties are needed. They are: the timeslot the class is scheduled for, the instructor teaching the class, and the classroom required for the class. The encoding used must be able to encode all the class properties that are required. The class properties are, the timeslot the course is scheduled for, the course which is going to be teached, the department of the course, the instructor teaching the course and the classroom for the course.

A timetable needs to be built around the following criteria : the rooms, instructors , class time, courses, and department.

The room class will contain information about the classroom, such as the roomID, room number and the capacity of students that can be accommodated. This class will accept a roomId, a room number and the capacity as well as provide methods to get the room’s properties.

The classTime class represents the day of the week and time that a class takes place. It contains the classTimeId and the classTime details.

The instructor class accepts a instructorID and instructor name properties. It also makes an allowance to retrieve this information as well.

A course class will store the information on the course modules. Each course can have number of students taking the course at different times of the week with different instructors. The course class accepts a courseID, departmentID, course name, an array of instructorID’s (instructor who teach the course).

A department class will store the information of all the courses which will be teched in the particular department. This class consists of departmentID and array of courses.

A termination check is set up such that the deciding factors are the number of generations and the fitness factor. Combining both of these factors will terminate the genetic algorithm either after a certain number of generations or if it finds a valid solution. As such the fitness value depends on the number of broken constraints. As a result, the perfect solution will have a fitness value of 1.

Uniform crossover is applied to guarantee that chromosomes are selected at random and are swapped with a parent, within the collections of genes.

Mutation is implemented in such a way that a new random valid individual is created. The random individual created is used to select genes to copy into the individual to be mutated. This is called uniform mutation. This technique ensures that all the mutated individuals are valid.

**RESULTS**

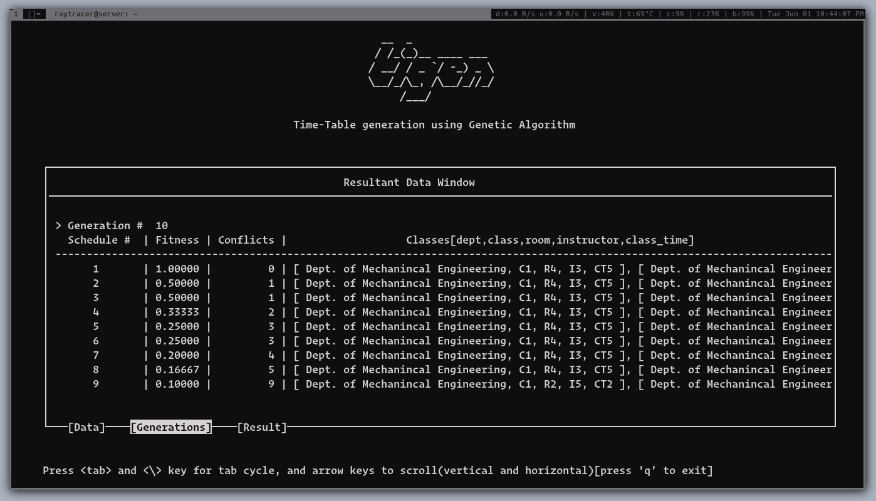
The Soft contrains and hard constraints were tested to ensure that all the solutions obtained were valid. The optimized solution for the timetable consisted of the following factors and their values. The population size was 100 with a mutation rate of 0.01%, a crossover rate of 0.9%, the number of crossover individuals was 2. With these values the resulting timetable had zero number of clashes with the fitness value of 1.

Figure 2: Generation

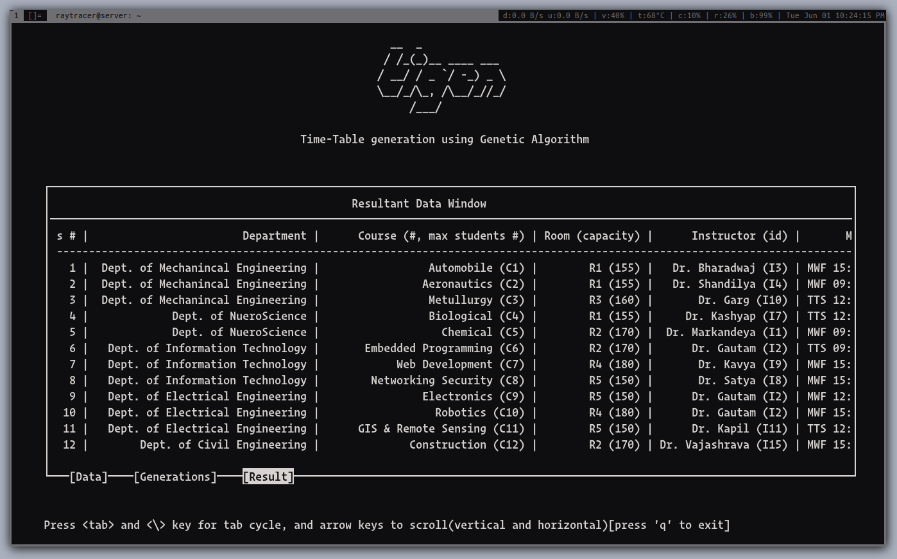
****

Figure 3 : Result TImetable

**ACKNOWLEDGEMENTS**

Foremost, We would like to express our thanks and sincere gratitude to our advisor Dr. Umesh Kumar Pandey for the unwavering support and encouragement. His patience, availability and vast knowledge in Genetic algorithm field helped us alot during this project. We would also like to thank Mr. Amit Kumar Tiwari whom we have also been blessed with the opportunity to learn and get advice from. his vast knowledge in this field have been very helpful in my project. We thank our department and the rest of our amazing faculty and staff. Thank you for your support and the wonderful job at the department. Finally, to our families, thank you for all your advice, sacrifice, encouragement and support throughout our life.

**CONCLUSION**

Scheduling is the very core working of any department or organization to perform well. TIGEN is the project which is going to provide a great help and benefit regarding this task. It will help in doing this important task as an automated process with very minimal manual interaction. So, in this era of increasing technology and automation, TIGEN is another automation tool and helper for human so that they can leave this important task to the project and just focus on other important aspect of their jobs and lives.

**REFERENCES**

* Sandeep Singh Rawat, Lakshmi Rajamani, “A Time table Prediction for Technical Educational System using Genetic Algorithm”, Journal of Theory
* MughdaKishorPatil, RakheShrutiSubodh, Prachi Ashok Pawar, NaveenaNarendrasinghTurkar, “Web Application for Automatic Time Table Generation”, International Journal of curre
* Pranavkhurana, “Timetable Schedular using genetic algortithm in JAVA”, Github
* Alberto colorni, Marco Dorigo, Vittoria Maniezzo, A GENETIC ALGORITHM TO SOLVE THE TIMETABLE PROBLEM, Centre for Emergent Computing , Napier University , Edinburgh EH10N 5DT, UK 2000.
* The course scheduler using genetic algorithms finds the best solution that satisfies a number of hard constraints. Also, the mutation technique used guarantees that the mutated chromosomes remain valid. This was done by creating a known valid random individual, swapping genes with it similar to uniform crossover. Use of uniform crossover and tournament selection completed the algorithm. The factors such as population size, mutation rate, crossover rate, elite individuals, and tournament size were considered for the course scheduler. The optimum values for these factors were obtained through several runs of the system. Also, the effect of mutation rate and the population size were studied for the course scheduler.

Before you begin to format your paper, first write and save the content as a

separate text file. Complete all content and organizational editing before

formatting. Please note sections \ref{AA}--\ref{SCM} below for more information on

proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been

formatted and styled. Do not number text heads---{\LaTeX} will do that

for you.

Define abbreviations and acronyms the first time they are used in the text,

even after they have been defined in the abstract. Abbreviations such as

IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use

abbreviations in the title or heads unless they are unavoidable.

\begin{itemize}

\item Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as ``3.5-inch disk drive''.

\item Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.

\item Do not mix complete spellings and abbreviations of units: ``Wb/m\textsuperscript{2}'' or ``webers per square meter'', not ``webers/m\textsuperscript{2}''. Spell out units when they appear in text: ``. . . a few henries'', not ``. . . a few H''.

\item Use a zero before decimal points: ``0.25'', not ``.25''. Use ``cm\textsuperscript{3}'', not ``cc''.)

\end{itemize}

Number equations consecutively. To make your

equations more compact, you may use the solidus (~/~), the exp function, or

appropriate exponents. Italicize Roman symbols for quantities and variables,

but not Greek symbols. Use a long dash rather than a hyphen for a minus

sign. Punctuate equations with commas or periods when they are part of a

sentence, as in:

\begin{equation}

a+b=\gamma\label{eq}

\end{equation}

Be sure that the

symbols in your equation have been defined before or immediately following

the equation. Use ``\eqref{eq}'', not ``Eq.~\eqref{eq}'' or ``equation \eqref{eq}'', except at

the beginning of a sentence: ``Equation \eqref{eq} is . . .''

\subsection{\LaTeX-Specific Advice}

Please use ``soft'' (e.g., \verb|\eqref{Eq}|) cross references instead

of ``hard'' references (e.g., \verb|(1)|). That will make it possible

to combine sections, add equations, or change the order of figures or

citations without having to go through the file line by line.

Please don't use the \verb|{eqnarray}| equation environment. Use

\verb|{align}| or \verb|{IEEEeqnarray}| instead. The \verb|{eqnarray}|

environment leaves unsightly spaces around relation symbols.

Please note that the \verb|{subequations}| environment in {\LaTeX}

will increment the main equation counter even when there are no

equation numbers displayed. If you forget that, you might write an

article in which the equation numbers skip from (17) to (20), causing

the copy editors to wonder if you've discovered a new method of

counting.

{\BibTeX} does not work by magic. It doesn't get the bibliographic

data from thin air but from .bib files. If you use {\BibTeX} to produce a

bibliography you must send the .bib files.

{\LaTeX} can't read your mind. If you assign the same label to a

subsubsection and a table, you might find that Table I has been cross

referenced as Table IV-B3.

{\LaTeX} does not have precognitive abilities. If you put a

\verb|\label| command before the command that updates the counter it's

supposed to be using, the label will pick up the last counter to be

cross referenced instead. In particular, a \verb|\label| command

should not go before the caption of a figure or a table.

Do not use \verb|\nonumber| inside the \verb|{array}| environment. It

will not stop equation numbers inside \verb|{array}| (there won't be

any anyway) and it might stop a wanted equation number in the

surrounding equation.

\subsection{Some Common Mistakes}\label{SCM}

\begin{itemize}

\item The word ``data'' is plural, not singular.

\item The subscript for the permeability of vacuum $\mu\_{0}$, and other common scientific constants, is zero with subscript formatting, not a lowercase letter ``o''.

\item In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)

\item A graph within a graph is an ``inset'', not an ``insert''. The word alternatively is preferred to the word ``alternately'' (unless you really mean something that alternates).

\item Do not use the word ``essentially'' to mean ``approximately'' or ``effectively''.

\item In your paper title, if the words ``that uses'' can accurately replace the word ``using'', capitalize the ``u''; if not, keep using lower-cased.

\item Be aware of the different meanings of the homophones ``affect'' and ``effect'', ``complement'' and ``compliment'', ``discreet'' and ``discrete'', ``principal'' and ``principle''.

\item Do not confuse ``imply'' and ``infer''.

\item The prefix ``non'' is not a word; it should be joined to the word it modifies, usually without a hyphen.

\item There is no period after the ``et'' in the Latin abbreviation ``et al.''.

\item The abbreviation ``i.e.'' means ``that is'', and the abbreviation ``e.g.'' means ``for example''.

\end{itemize}

An excellent style manual for science writers is \cite{b7}.

\subsection{Authors and Affiliations}

\textbf{The class file is designed for, but not limited to, six authors.} A

minimum of one author is required for all conference articles. Author names

should be listed starting from left to right and then moving down to the

next line. This is the author sequence that will be used in future citations

and by indexing services. Names should not be listed in columns nor group by

affiliation. Please keep your affiliations as succinct as possible (for

example, do not differentiate among departments of the same organization).

\subsection{Identify the Headings}

Headings, or heads, are organizational devices that guide the reader through

your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not

topically subordinate to each other. Examples include Acknowledgments and

References and, for these, the correct style to use is ``Heading 5''. Use

``figure caption'' for your Figure captions, and ``table head'' for your

table title. Run-in heads, such as ``Abstract'', will require you to apply a

style (in this case, italic) in addition to the style provided by the drop

down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For

example, the paper title is the primary text head because all subsequent

material relates and elaborates on this one topic. If there are two or more

sub-topics, the next level head (uppercase Roman numerals) should be used

and, conversely, if there are not at least two sub-topics, then no subheads

should be introduced.

\subsection{Figures and Tables}

\paragraph{Positioning Figures and Tables} Place figures and tables at the top and

bottom of columns. Avoid placing them in the middle of columns. Large

figures and tables may span across both columns. Figure captions should be

below the figures; table heads should appear above the tables. Insert

figures and tables after they are cited in the text. Use the abbreviation

``Fig.~\ref{fig}'', even at the beginning of a sentence.

\begin{table}[htbp]

\caption{Table Type Styles}

\begin{center}

\begin{tabular}{|c|c|c|c|}

\hline

\textbf{Table}&\multicolumn{3}{|c|}{\textbf{Table Column Head}} \\

\cline{2-4}

\textbf{Head} & \textbf{\textit{Table column subhead}}& \textbf{\textit{Subhead}}& \textbf{\textit{Subhead}} \\

\hline

copy& More table copy$^{\mathrm{a}}$& & \\

\hline

\multicolumn{4}{l}{$^{\mathrm{a}}$Sample of a Table footnote.}

\end{tabular}

\label{tab1}

\end{center}

\end{table}

\begin{figure}[htbp]

\centerline{\includegraphics{fig1.png}}

\caption{Example of a figure caption.}

\label{fig}

\end{figure}

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words

rather than symbols or abbreviations when writing Figure axis labels to

avoid confusing the reader. As an example, write the quantity

``Magnetization'', or ``Magnetization, M'', not just ``M''. If including

units in the label, present them within parentheses. Do not label axes only

with units. In the example, write ``Magnetization (A/m)'' or ``Magnetization

\{A[m(1)]\}'', not just ``A/m''. Do not label axes with a ratio of

quantities and units. For example, write ``Temperature (K)'', not

``Temperature/K''.

\section\*{Acknowledgment}

The preferred spelling of the word ``acknowledgment'' in America is without

an ``e'' after the ``g''. Avoid the stilted expression ``one of us (R. B.

G.) thanks $\ldots$''. Instead, try ``R. B. G. thanks$\ldots$''. Put sponsor

acknowledgments in the unnumbered footnote on the first page.

\section\*{References}

Please number citations consecutively within brackets \cite{b1}. The

sentence punctuation follows the bracket \cite{b2}. Refer simply to the reference

number, as in \cite{b3}---do not use ``Ref. \cite{b3}'' or ``reference \cite{b3}'' except at

the beginning of a sentence: ``Reference \cite{b3} was the first $\ldots$''

Number footnotes separately in superscripts. Place the actual footnote at

the bottom of the column in which it was cited. Do not put footnotes in the

abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use

``et al.''. Papers that have not been published, even if they have been

submitted for publication, should be cited as ``unpublished'' \cite{b4}. Papers

that have been accepted for publication should be cited as ``in press'' \cite{b5}.

Capitalize only the first word in a paper title, except for proper nouns and

element symbols.

For papers published in translation journals, please give the English

citation first, followed by the original foreign-language citation \cite{b6}.