THE UNIVERSITY OF Nottingham

School of Computer Science and Information Technology

An Automated Space Allocation System – Technical Documentation

D.B. Varley
dbv@cs.nott.ac.uk

Automated Scheduling And Planning Group
Department of Computer Science
University of Nottingham
Nottingham, UK.
NG7 2RD.

Table of Contents

1. INTRODUCTION

This report documents the technical details and specification of the Space Allocation system which has been developed under the JISC project Automated University Space Allocation by the Automated Scheduling and Planning group of the University of Nottingham. This report in combination with the companion User Manual will provide the documentation for the system. The User Manual will report the systems interface and the usage of the system whereas this report will document the underlying algorithms, data structures and applicable information regarding the problem of space allocation.

With the introduction of modularity, increasing student numbers and the continued expansion of university departments, space is becoming an increasingly precious commodity. Continued attempts to alleviate this problem have been made, all intending to increase the efficiency of how space is utilised.

Some of these attempts have included changes to what decisions are made when space is allocated and how the allocation process makes use of the available tools. These tools vary from the introduction of computerised software to store details of rooms, such as databases and Computer Aided Drawing (CAD) to new methods of monitoring space utilisation such as Space Charging.

As this process of space allocation varies considerably between different HE institutions, a single, complete solution to this problem is virtually impossible. However, there is scope for the automation of parts/or the majority of this process, through the use of meta-heuristic methods.

The Automated Scheduling and Planning group saw such possibility for automation and have created a system to provide a useful tool for space administrators.

2. PROBLEM DEFINITION

A basic definition of what the space allocation process involves, is desirable:

> *"The allocation of resources and/or functionality to areas of space such as rooms"*

Obviously, it is also desirable to perform this process as efficiently as possible, therefore a further definition would be:

> *"The allocation of resources and/or functionality to areas of space such as rooms, satisfying as many second order constraints as possible without violating any first order constraints"*

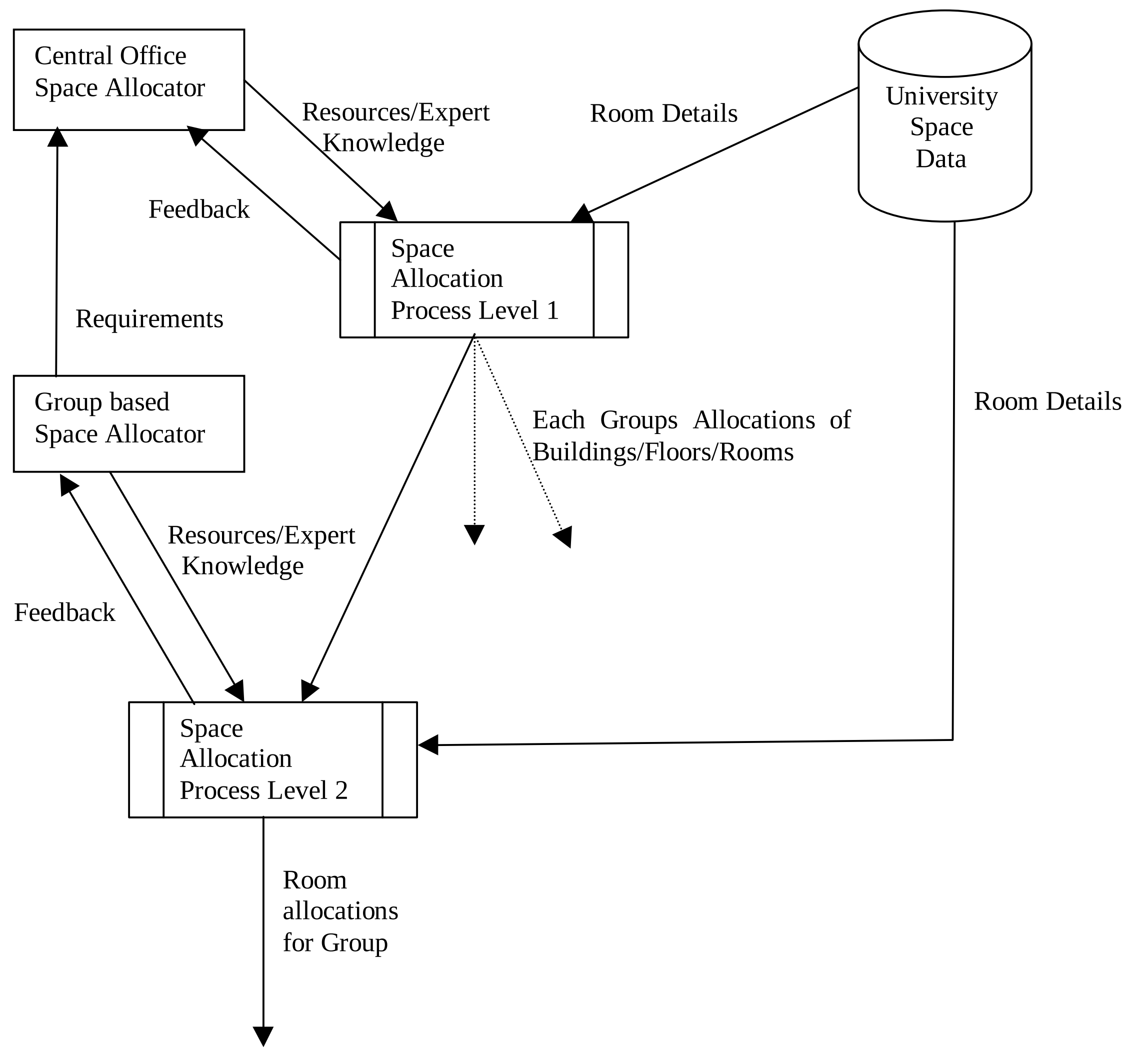
First order constraints represent the rules of each institution which cannon be violated in any way. Second order constraints are the rules which can be broken, but an appropriately weighted penalty is applied. By minimising the penalties applied during the space allocation process and ensuring no violations of first order constraints, the most efficient allocation of space can be performed.

2.1 Model of the Problem

An example of the typical space allocation process within British Universities is given below.

This example shows the space allocation process as two distinct sub-processes. The first (level 1) being the splitting up of buildings/floors/rooms by the Central Office for each Faculty/School/Department. Level 2 is the process of each department allocating resources/functionality to the rooms they have been given.

Simpler models are available, as are far more complex ones, all dependent on the methods used within each University. This example simply aims to show the main principle behind space allocation.



2.2. Applying Space Allocation

The model of the problem shown above emphasises that the process of Space Allocation is a multi-stage problem. Different levels of authority within a university have different tasks within Space Allocation. Higher level authorities have to view the problem of space in a wide view encompassing all lower levels. As a result of this, they cannot be expected to consider actual individual resource to individual room allocations. They consider the *big picture* and allocate lower level groups to specific buildings/floors, based on information provided by the groups. This leaves the groups to then allocate their own resources and request more space if necessary.

2.2.1. Departmental Space Allocation

Space Allocation is not just a single process, which is run maybe once a year. Regardless of the data collection and analysis stages, the actual allocation of resources to rooms can be applied in four different ways, some used more frequently than others:

- **Fitting all resources into a limited number of rooms**

  A department is allocated a certain number of rooms and they must allocate their resources to these rooms, fitting in all (or the majority) in the most efficient manner.

- **Minimising the number of rooms required for a set of resources**

A department needs to be located within a building and they must calculate how many rooms they need to fully allocate all their requirements and resources.

- **Adding/removing rooms or resources from a current allocation**

  Resources or rooms must be added or removed to what is currently available in the best possible way, without causing too much reorganising. A balance must be made between ensuring efficient utilisation and not causing too much disruption to the resources already allocated.

- **Reorganising/optimising the current allocation**

  The department realises that it's current allocation is inefficient and wishes to take the opportunity to analyse the current allocation and where possible, optimise it by moving resources around within the rooms that are available.

The first two methods are used in situations where a department moves into a new building or office space. They look at the same problem from different viewpoints. The first asks how many of the resources can be placed within a pre-allocated finite number or rooms. The second asks how many rooms do we need ideally? The third option is most likely to be used frequently, allowing small changes to be made to an existing allocation while maintaining efficient utilisation. The last option is for departments, which know they have poor space utilisation and wish to reorganise, regardless of the cost in moving and disruption. This option is unlikely to be used unless absolutely necessary.

2.3 Data Requirements for Space Allocation

The Space Allocation process involves making a huge amount of decisions on the applicability of allocations between rooms and resources. Decisions on whether one resource would better use a room are common and allow the optimal usage to be found. Obviously, these decisions require information or data to base the conclusions upon. The data requirements for the Space Allocation process vary from the obvious need for a list of rooms and resources, to extra data to enable additional considerations to be included.

2.3.1 Space Utilisation Based Allocation

To make a low level space allocation based solely on Space Utilisation the following data is required:

- **Resource Lists**

  List of all resources requiring rooms. Must include data on resource type (i.e. Staff, Student, Lecture Room, Laboratory, etc.) Personnel names can be omitted to protect privacy.

ID	***LEVEL***	***QUANTITY***
Professor #1	*Professor*	*1*
Lecturer #3	*Senior Lecturer*	*1*
Small Lecture Room	*Lecture Room*	*20*
Main Computing Lab	*Computing Laboratory*	*60*
FIB Group Researcher #1	*Researcher*	*1*

- **Room Lists**

  List of all rooms available for allocation. Must include size/capacity information.

ID	*SIZE (m^2)*
1105	*152.4*
A13	*34.2*
Lecture Room #1	*64.5*

- **Space Standards**

  List of standard size requirements for each type of resource requiring allocation

LEVEL	*SIZE REQUIREMENT (m^2) PER OCCUPENT*
Professor	*12.0*
Senior Lecturer	*9.0*
Computing Laboratory	*3.0*
Lecture Room	*1.0*

2.3.2 Additional Constraint Based Allocation

To include constraint satisfaction considerations to the Space Allocation process, far more information is required. The information required is applicable to the constraint requiring satisfaction, e.g:

- **Proximity Constraints**

  Constraints involving allocating n resources adjacent or within close proximity to each other require details as to the location and proximity of each room to each other room.

ID	*SIZE (m^2)*	*BUILDING*	*FLOOR*	*ADJACENT ROOMS*
1105	*152.4*	*Tower Block*	*11th*	*1104, 1106*
A13	*34.2*	*Pope*	*Ground*	*A12*
Lecture Room #1	*64.5*	*Trent*	*1st*	

- **Special Requirements Constraints**

  Constraints involving special requirements such as disabled access require that all the rooms which cater for these special requirements have extra information.

ID	*SIZE (m^2)*	*DISABLED ACCESS*	*AVA FITTED*
1105	*152.4*	*No*	*No*
A13	*34.2*	*Yes*	*No*
Lecture Room #1	*64.5*	*Yes*	*Yes*

- **Room Sharing Constraints**

  Constraint restricting certain resources from sharing rooms obviously requires the information which tells us which resources can or cannot share. The resource lists must be updated with departmental and/or group information so we know who is with which group.

ID	*LEVEL*	*QUANTITY*	*DEPARTMENT*	*GROUP*
Professor #1	*Professor*	*1*	*Maths*	
Lecturer #3	*Senior Lecturer*	*1*	*Maths*	
Small Lecture Room	*Lecture Room*	*20*	*Physics*	
FIB Researcher #1	*Researcher*	*1*	*Computing*	*FIB*
Main Computing Lab	*Computing Laboratory*	*60*	*Computing*	

LEVEL	*SHARING CAPABILITIES*
Professor	*Don't share*
Senior Lecturer	*Don't share*
Researchers	*Share with same level & group*
Secretaries	*Share with same level*

The constraints listed above represent the main ones involved in Space Allocation. A detailed analysis of all the constraints which concern Space Allocation are included in section 3. Each of these constraints will most likely require information which is either available or not. A balance must be made between the cost of obtaining this information against the benefits of using these constraints in the process.

3. CONSTRAINTS AND CONSIDERATIONS

The following sections represent a list of all constraints, variations and considerations, which affect the space allocation process. The items within this list have been gathered through various sources, mainly an analysis of a questionnaire [1] that was sent to all HE institutions within the country. Although every attempt has been made to ensure a nearly exhaustive list, the lack of previous work in this problem field limits the availability of information.

3.1. Ownership of Space

One of the most varied aspects of space allocation within HE is the control and ownership of space. Within different institutions, there is a hierarchy of possible owners:

Externally Controlled
For certain exams/lectures the institution may have to hire external rooms due to a variety of reasons. These may be that the institution does not have an adequate room or those rooms are full, etc. These rooms are more used in timetabling/scheduling but the space allocators must know of their existence.

Centrally Controlled
The institution has an estate office or centrally controlled department, which allocates room to the separate groups within the institution (departments, faculties, schools, etc.)

Faculty/School/College Controlled
Each large group within the institution is allocated specific building(s) or floor(s) within buildings and are allowed to allocate this space for themselves or their internal departments at their own discretion.

Departmental Controlled
Each department is allocated specific building(s) or floor(s) by either the faculty or the estate office and is allowed to allocate this space at their own discretion.

Variations also occur with the different requirements for rooms, i.e. Rooms that act as office space for academic, administrative staff or research students, are usually allocated by the department. Whereas, large lecture rooms or rooms used for examinations are usually centrally allocated. However, some departments also insist on having sufficient lecture space departmentally controlled.

Due to the fact that a potentially large number of controllers are possible within a large institution, any system to allow control of all space allocation must be capable of sharing information over a large area. Institutions with multi-site campuses will inevitably wish to produce complete reports of all buildings that would not be possible if the data was kept in many separate segments.

3.2. Room Requirements

The various functionality's that can be allocated to rooms is usually common amongst institutions, however some variations will exist. There are approximately 30 different functionality's which are used but the most common types are:

Staff Offices
Used for accommodating academic, administrative or technical members of staff.

Research Offices
Used for accommodating research groups and postgraduate students.

Storage/Equipment/Administrative
Used for storing photocopiers, stationary, computer servers, etc.

Library Space
Used for the storage and consultation of books, academic papers, etc.

Recreational/Amenity
Staff and common rooms, cloakrooms, toilets, wash rooms, etc.

Lecture Theatres
Used for giving lectures and presentations, usually large and requiring special resources.

Meeting Rooms
Used for holding group discussions, tutorials, small presentations, etc.

Laboratories
Dependant of type of laboratory and department but usually require special resources.

How the rooms are to be used usually specifies the size requirements. Different HE institutions use differing guidelines towards how much space to allocate for each person or equipment within each room. An example of these guidelines is the Full Time Education (FTE) 1987 Space Standards, a part of which is shown below:

Types of room	Area in square metres
Professors, including small group space	18.5 per room
Tutorial staff, including small group space	13.5 per room
Non-tutorial staff (Researchers, Secretaries)	7.0 per person
Work Study Space for Postgraduate	3.4 per person
Drawing Offices A1 boards	3.7 per place
Computing Laboratory	5.0 per place
Lecture Theatres with close seating	1.0 per place

Other guidelines such as PCFC and HECFE are also used. Some institutions have created working parties to create their own guidelines and some use no guidelines at all, improvising when required. Regardless of where they come from, these size guidelines are instrumental in creating an automated process.

3.3. Room Groupings

3.3.1 Departmental Based Groupings

All institutions prefer and usually insist that rooms allocated to the same department/faculty/school/college are located close to one another. The level of closeness depends on the size of the group but often, complete buildings are allocated to single or related groups. Where space is limited or groups are small, different groups may be allocated separate floors within shared buildings, however, sometimes even floors are shared between groups.

3.3.2 Specific Functionality Groupings

Some institutions express a requirement to ensure certain allocated rooms have a close proximity to other rooms. For example, Departmental Secretaries being close to Heads of Departments, Research Leaders being close to their Research Groups, etc. Departments may also require that all the lecture and meeting rooms are located closely to each other, or that all staff offices are on the same floor.

There are two levels of grouping normally required, adjacency and proximity. Adjacency groupings require that resources be located next to each other (e.g. Scrub Rooms and Operating Theatres). Adjacency groupings sometimes require special access between rooms such as adjoining doors, etc. Proximity groupings are less strict and simply require that resources be located within a close proximity (e.g. within one or two rooms, same floor, same building, etc.). Regardless of which type of grouping is required, different weightings can be applied to each grouping so that some are considered more important and therefore stricter than others.

3.4. Room Capacities

Each room will have a size value which when compared against the space standards for that room type will give its capacity. The size value, usually measured in square metres, will either be a hand measurement or calculated from a Computer Aided Design (CAD) tool. To ensure optimal usage of space, it is desirable to maximise the proportion of the room capacity used. These capacities will limit the functionality, which can be allocated to specific rooms, (i.e. a 10 square metres room is unlikely to be used as a lecture theatre (10 student capacity)).

3.5. Building Work

Sometimes a room of the size required is not available anywhere and it is therefore required to perform building work. This could include removing a wall to make two small rooms into one large one or vice-versa. This is often used only as a last resort, however the possibility exists and therefore the option must be available.

Other building work such as creating an adjoining door or window between two rooms is sometimes required to increase utilisation.

3.6. Sharing Rooms

Dependant on the institution and the level of the people being allocated rooms, sometimes it is required to make people to share rooms. Likewise, it is also required to ensure that certain people *do not* share rooms. A common setting is that academic staff do not share rooms, whereas research group members and postgraduate students are expected too. Regardless of whether sharing is forced or not, certain academic/technical members of staff may need a single room due to security or confidentiality reasons. Once again, consideration of the room's capacity must be given.

3.7. Room Usage/Utilisation

Once a room has been allocated to a specific purpose, an analysis of how often the room is utilised is useful in monitoring the overall efficiency. This measurement is especially useful for rooms allocated to laboratory, lecture, and examination or meeting purposes. Rooms, which are infrequently used, may be more effectively used for another function. This aspect of space allocation ties closely with the timetabling process. An investigation of the dividing line between space allocation and timetabling and an analysis of the sharing of information between the two processes is required.

Note: Efficient space utilisation requires both good quality space allocation and good quality timetabling.

3.8. Monitoring Space Details

However the space is allocated or by whom, there is always a need to keep track and to monitor the current allocation details. Whether this is to produce annual reports on how much space each group has, or whether it is for making decisions on whether more space is needed. This is a major factor of space allocation as many of the new methods of ensuring utilisation is based on a firm and detailed database. Much of this information must be provided from the room scheduling or timetabling process, therefore, inter-process exchange of information is essential.

3.8.1. Space Charging

Of the various methods to ensure efficient space usage, one of the newest and increasingly popular is Space Charging. This method uses utilisation figures for space usage in departments and charges those departments who are not making efficient use of their space. These figures also enable a decision to be made when extra space is requested. Departments, who use their current space poorly, will not be allocated extra space when requested.

3.9. Global or Departmental Facilities

Most HE institutions are split between the need for having sufficient laboratory, lecture and library space within each department or having centralised facilities used by all departments. The former situation requires more space and the latter requires more management/timetabling. Usually a dividing line is found where departments have a small amount of departmental facilities, but rely on global facilities for larger groupings.

3.10. Special Allocation Requirements

To allocate a specific functionality to a room sometimes addition requirements must be met. Lecture/examination rooms may need to have disabled access, or Audio Visual Aid (AVA) facilities, etc. Library space may need to be located in a quiet area away from busy rooms, noisy equipment. Laboratories may need extra protective glass, walls, etc.
Such information must be available to judge whether addition cost/work must be committed before allocation.

3.11. Reorganising

There is a major difference in the requirements for space allocation between initially allocating functionality to *empty* rooms and reorganising previous allocated rooms. One of the major constraints with reorganising is minimising the amount of disruption that it causes. This constraint often impedes finding the optimal utilisation of space due to the fact that it is too costly to completely move everyone around. The quality of the initial allocation usually defines how much reorganising is required at a later date. An optimal initial allocation will need little changing when resources are added or removed.

Continual reorganisations on a small scale usually result in a bad overall utilisation of space, however large reorganisations are time consuming and costly. The limit to how much disruption should be allowed must be controlled to balance the quality of the new allocation to the difficulty in achieving it.

4. SYSTEM STRUCTURE

The following section will discuss and document the data structures and specifics of the algorithms and coding used within the Space Allocation system. This is to enable users to gain an understanding of the inner workings of the system and also for future modifications to be made by referencing this document.

The system has been designed and created for use on PC's running 32 bit Windows environments such as Windows 95/98 and Windows NT. The decision to develop the system for IBM PC's and 32 bit Windows was made on the results of the questionnaire[1], which showed that the majority of British universities are running this configuration.
The system ahs be programmed using the Microsoft Visual C++ development package currently at version 6.0. It makes full use of the library functions available within Windows such as Open DataBase Connectivity (ODBC) and Object Linking and Embedding (OLE) to make the system as usable, informative and intuitive as possible.
The system is split into two main components, the algorithms and underlying data structures and the main user-interface. The interface will be discussed thoroughly in the User Manual report which should come with this report. Therefore we will specifically deal with the algorithms and underlying data structures in this report.

Data is the foundation in which any decision based algorithm or process is based. The Automated Space Allocation system is no different. Without data, the results of the system will be meaningless. The better quality data the better the result will be. Therefore, it is essential to ensure that all users can use, modify and easily work with their own data.

4.1 Ensuring ODBC Is Set Up

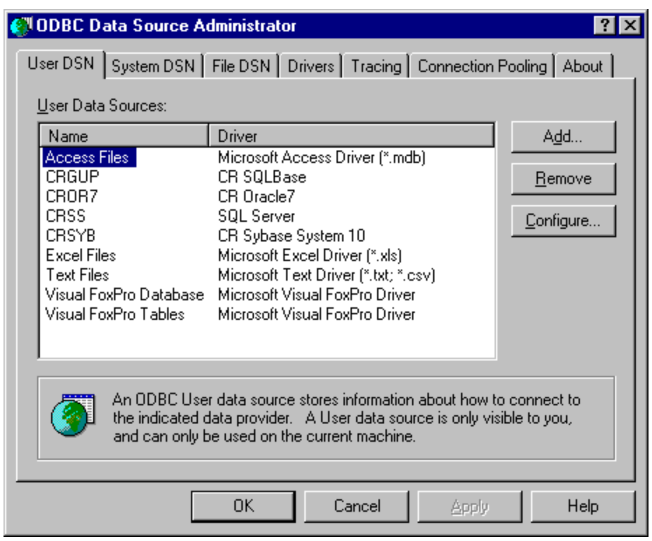
The ODBC capabilities of Windows are included as default with the operating system in a limited configuration. Additional drivers for database formats are available and included with software such as Microsoft Office, which uses Access as the main database tool. The Space Allocation system will make use of whichever drivers are installed, however if you wish to use a specific type of database file, you must ensure that the appropriate ODBC driver is installed.

This is reasonable simple as if the software package you are using for the database file is installed, such as FoxPro, Access, then the ODBC drivers would have been installed by the software.

This section is simply to enable users to confirm that their ODBC drivers are set up correctly to eliminate the possibility of problems occurring with use of the Space Allocation system.

The procedure for confirming your ODBC drivers is as follows: -

- `Select Settings->Control Panel from the Windows Start Menu`
- `From the Control Panel, select the icon called either ODBC Data Sources or ODBC 32 bit`
- `Now your should see the ODBC Administrator window which will tell you which data sources have been set up.`



- `Ensure there are data sources set up for the database types you expect to be using.`
- `If the appropriate data sources are not set up, re-install the database software you use or if you are experienced in setting up data sources use the Add button.`
- `The two main data sources should be Text Files->Microsoft Text Driver(*.txt;*.csv) and Access Files->Microsoft Access(*.mdb)`

4.2 Input Data Format

It is important that all users can (easily and with the minimum amount of work) use their own data with the Space Allocation system. This is a major concern as virtually all British Universities have a huge store of space allocation information stored on computers in various formats. There are many different formats for data transfer, many of which are virtually obsolete. Data may come from PC's, UNIX Workstations, IBM Mainframes and as such, the format used for the Automated Space Allocation system must be compatible or easily convertible to all these systems. The format decided upon based on information from the MAC Initiative(and other sources, was Text based Command Separated Values (CSV) format. This format is standard amongst all operating systems, easy to convert to/from standard SQL, Ingres and Oracle database files. It is also the format by which most HEFCE planning data is stored. The only concern with this format is that, as it is text based, the level of security is virtually non-existent. However, for those users concerned about security, limiting access to the computer, the drive or directory, which the files are stored in, should be sufficient. At the extreme, encryption can be used on the files to provide password based file access.

The Windows environment accesses database files such as SQL, Oracle and Text CSV using the Open DataBase Connectivity (ODBC) libraries. These libraries allow conversion of differing formats easily and also allow general reading of most standard formats. This means that the Automated Space Allocation System should be able to read database files in any format which ODBC supports, however, starting with CSV ensures that at least one format is 100% supported.

An example of the data format is shown below:

```
“ID”,”SIZE”,”FLOOR”,”BUILDING”,”ADJACENT ROOMS”
"1101",13.20,"11th Floor","Tower Block","1102,1103"
"1102",10.80,"11th Floor","Tower Block","1101,1103"
"1103",9.40,"11th Floor","Tower Block","1101,1102,1104"
```

The data can also be filtered so that only the data required by the space allocation process at any time is read into memory. This ensures that no excessive memory requirements are needed to run the system. For example, a user wishing to perform space allocation on one building on campus will only read in the data for that building, not the other 50 or so buildings as well. This is quite essential when it is considered that the rooms database for the University of Nottingham is 5.2Mb in size.

4.3 Input Data Structure

The data is read into the system using the principle of file sets. A file set is a set of files with each different file holding the CSV data for a certain type of data. The initial design of the system has categorised the file set into the following groups:

- **Resources**
  Lists of all resources within the organisation with their appropriate information.

- **Rooms**
  Lists of all rooms requiring allocating to rooms within the organisation.

- **Constraints**
  Lists of the standard and user defined constraints used in the space allocation process.

- **Guidelines**
  Lists of space requirements and guidelines for each type of resource to be allocated. Can also include factors allowing compromises on some guidelines.

- **Allocations**
  Lists of resource to room allocations which already exist before the system is run. This can be a previous set of allocations created by the system or can be real-world allocations.

This principle of file sets allows the user to vary one of these categories while keeping the others constant. For example, if all this data were kept in a single file, to change the guidelines would involve copying the whole file and then changing the small area, which listed the guidelines. This would result in two large files with mainly the same data. The problems with data redundancy and scope for errors are tremendous. This way, simply the single file with the guidelines must be copied altered and changed in the file set.

These five groups cover the whole scope of information, which needs to be supplied to a space allocation system. Guidelines could be grouped together with Constraints, but as the guidelines are inherent and essential to the process, they have been separated. They still qualify as hard constraints but their importance justifies the separation.

The file sets are read into the system using the standard Windows menu options and dialog boxes.

4.3.1 Resource Data Structure

The following information is for the structure of the resource data group of the file set. Certain fields are essential and must be included for the Space Allocation system to use this data.

NAME (essential)

The name or unique ID or each resource, specific person names can be omitted for privacy but an informative unique identifier is required for the system to distinguish between similar level resources.

LEVEL (essential)

The level of the individual resource, this must match the levels listed with the guidelines data group so the system can calculate how much space the individual resource requires.

DEPT

Optional field to store information about the department the resource belongs to.

GROUP

Optional field to store information about any research/technical or departmental groupings which may be applicable.

USE

Optional field for those administrators who store information about resources which have left or are no longer applicable. This field allows the user to specify to the system whether the resource is to be included (1) or ignored (0) when reading in the resource data.

SHARE

This field allows the user to overide the general sharing constraints specified in the requirements data group. This is useful for those resources which go against their levels default sharing requirements, e.g. a Part-time lecturer who should share a room, although lecturer generally don’t.

4.3.2 Rooms Data Structure

The following information is for the structure of the room data group of the file set. Certain fields are essential and must be included for the Space Allocation system to use this data.

ID (essential)

A unique identifier for each room which can be as informative as you like.

SIZE (essential)

The size of the individual room in square metres.

BUILDING

Optional field specifying which building the individual room is located within.

FLOOR

Optional field specifying which floor of the building the room is located on.

ADJACENT

A list of optional rooms which the current individual room is located adjacent too. This is used by the algorithms to calculate a rough plan of where each room is located relative to others. Is used for adjacency and proximity constraints.

4.3.3 Constraint Data Structure

The following information is for the structure of the constraint data group of the file set. All fields are essential and must be included for the Space Allocation system to use this data, although some can be left blank.

LABEL

Unique label defining the constraint, so that constraint can be combined in (if..then..else), (and/or) type combinations.

SUBJECT_LIST

List of specifications for the type of resource/room which the constraints applies to.

e.g. *Resource->Id* = “E Burke” and *Resource->Level* = “Lecturer”

Note: Capable of specifying ALL, AND, OR, towards any values to any column in any table.

CONSTRAINT

A specific constraint or relationship between the Subjects and Targets of the constraint.

e.g. *located near, located next to, have, located in, share with,* etc.

NEGATION

Applies to the CONSTRAINT field in that this allows negation of the constraint.

e.g. *located near* would be negated to, *not be located near.*

TARGET_LIST

Same as SUBJECT_LIST except this lists the specifications for the constraint.

Note: Capable of all SUBJECT_LIST requirements and be able to specify NULL or empty sets.

TYPE

Type of CONSTRAINT, whether *Hard* or *Soft*. Effectively adds *Must* or *Should* to the full constraint.

EXPONENT

Exponent part of the fitness penalty to apply for every occasion in which the constraint is violated. The fitness penalty can be made up of an exponent part and/or a weighting.

WEIGHTING

Weighting part of the fitness penalty to apply for every occasion in which the constraint is violated. Without an exponent part, the weighting acts as a simple factor.

4.3.4 Requirements Data Structure

The following information is for the structure of the requirements data group of the file set. Certain fields are essential and must be included for the Space Allocation system to use this data.

LEVEL (essential)

Resource level equivalent to the similar field in the resource data structure. This matching is used to join the two data structures.

SIZE (essential)

The size requirement for a single instance of the individual level, in the case of staff etc. this represents a single Professor/Lecturer, however in the case of multiple capacity resources such as Lecture Rooms this figure represents the size requirements for a single occupant of the room.

SHARING

This represents the standard sharing requirement for the individual resource level. It can be overridden in the resource data structure on an individual resource basis.

4.3.5 Allocations Data Structure

The following information is for the structure of the allocations data group of the file set. Certain fields are essential and must be included for the Space Allocation system to use this data.

ROOM (essential)

The individual room identifier, matching a room within the room data group which has resources allocated to it.

ALLOC (essential)

A list of resource names, matching those within the resource data group, to allocate to the individual room.

4.4 Algorithm Information

This section will discuss the algorithms available within the Space Allocation system, their capabilities and differences.

Two different meta-heuristic methods are used in the algorithms, the first being Hill-Climbing and the second being Simulated Annealing. Further descriptions of the academic aspects of these methods can be obtained with the author's academic papers referenced in this report.

To the user, the two differing methods can be classified as quick and intensive. The first algorithm (Hill Climbing) represents a quick and fast way of checking whether the allocation of space can be improved in any way easily. This method will not spend lots of time searching for small improvements but will come up with any obvious or not-so obvious changes. The second algorithm (Simulated Annealing) will search more thoroughly to try and find the best possible solution to the current allocation problem.

Two algorithmic methods have been included so that the users can choose between finding a quick (minutes) solution or an intensive (15 minutes +) solution. Obviously, the longer the algorithms are left to run, the more likely they are to find a better solution, generally.

Both algorithms make decisions based on possible movements evaluated relative to the constraints and requirements of the space allocation problem. These decisions are reviewed in greater detail in reference 2, as well as various analyses of the performance of the methods, however, a short description is shown here.

The possible moves of the algorithm are: -

- Moving an individual resource from one room to another, the unallocated bin counting as a room,
- Swapping all resource from one room with all resources in another.

The moves effectively allow for small or large changes in the allocation. It should be noted though, that all moves are performed dependant on the evaluation of the move, therefore if it is a bad move to swap six people from one room with 1 person from another, then the algorithm will not make the move.

4.4.1 Algorithm Evaluation

In order to ascertain the quality of a space allocation solution, a measure of the overall resource allocation, space utilisation and constraint satisfaction is needed. The following equation represents a generalised penalty function for any algorithmic method:

$$Penalty = \mathrm{Re}\,sourcesUnscheduled + \sum_{i=0}^{NoOfRooms-1} \left(Wastage(i) + SpacePenalties(i) \right)$$
$$+ \sum_{x=0}^{NoOf\,\mathrm{Re}\,sources-1} \; \sum_{y=0}^{NoOf\,\mathrm{Re}\,sources-1} \mathrm{Re}\,sourceConflicts(x, y)$$

Applying weights allows certain constraints to be considered to be more important than others and therefore have differing penalties associated with them. These weights are represented with the system as constraints, input by the user within the constraints data group. The weights shown below represent defaults used within the testing and research parts of the project, for each of the sections of the equation above. Users can change and modify these figures any time.

Each constraint has an exponent and factorial weighting allowing greater versatility in applying penalties. An exponent of one represents a consistent penalty, i.e. each resource that is unscheduled increases the penalty by 5000. An exponent of greater than one represents an increasing exponential weighting depending on the size of the violation, i.e. exceeding room capacity by 2.0m^2 increases the penalty by 4, exceeding by 15.0m^2 increases the penalty by 225.

Constraint	**Exponential**	**Factor**
Resources Unscheduled – resources not allocated to rooms	1.0	5000.0
Space Wastage (per m^2) – not using full capacity of room	1.0	2.0
Space Penalties (per m^2) – exceeding room capacity	2.0	2.0
Sharing Violations – resources sharing when not allowed	1.0	2000.0
Grouping Violations – members of different groups sharing	1.0	1000.0
Adjacency Requirements – resources requiring adjacent placing	1.0	500.0
Grouping Requirements – resources requiring same/adjacent room placing	1.2	50.0
Proximity Requirements – resources requiring proximity placing	1.0	750.0

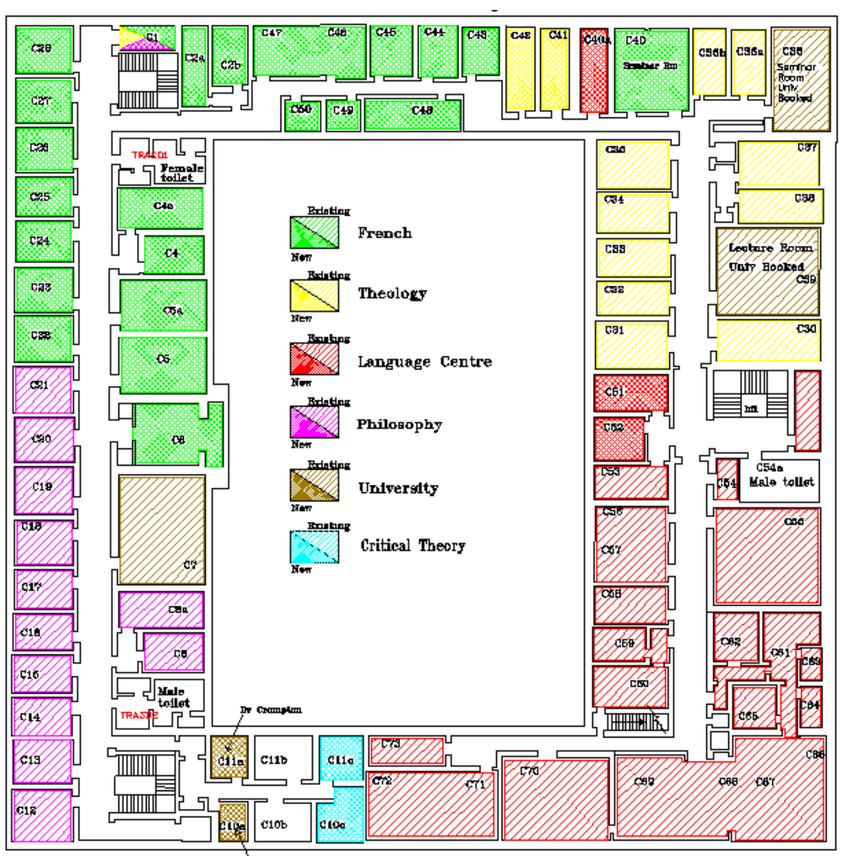
The evaluation function of the algorithms is the most time consuming and important part of the Space Allocation system, it gives the user a mathematical representation of how good the solution shown within the system is. However, it is obviously dependent on the constraints, weightings and input data provided by the user.

4.5 Floor Plans Information

The Space Allocation system can display CAD drawings to show the floor plan layout of allocations if those files are available. However, this functionality has not been perfected for all drawings types. This section will list the limitations and capabilities of this aspect of the system.

Computer Aided Design (CAD) drawings are used throughout most universities to keep plans of the building structures and room layouts of all buildings. These drawings are to-scale representations and can be used to obtain much useful knowledge about rooms, etc. However, the capabilities of CAD allow for a myriad of possible drawing type from simple pictures of floor plans, three-dimensional representations of buildings to object related room plans. These various possibilities make it almost impossible to configure a system to display CAD floor plans. The Space Allocation system can, at the moment, display 2D Floor Plans in most Windows supported picture formats, BMP, CAD, GIF, etc.

The links between rooms and floor plans at the moment is made through the resource data group, an additional column is used to store the filename of the appropriate plan to display when requested. This must be done on a room rather than a floor basis, because some floor plans are too big to adequately show enough information about individual rooms. The usage of these floor plans is optional and completely up to the users desecration.



The above picture shows an example of the kind of information available with CAD Floor plans. The system is currently being modified to enable the floor plans to be modified to represent the current solution within the system. Therefore, clicking on a room would display the occupants and room sizes, etc.

4.6 Graphical Statistics Information

The Space Allocation system can also graphically show statistical information about the current allocation. These graphs are available to be viewed, printed etc. Only a few graph formats are available at the moment, but dependant on the requests and information gained from testers/users of the system, more formats can be added at a later date.

4.7 System Source Code Information

Due to various requests for the source code for the system, the following section has been added.

The system is available freely to any university interested, however the source code is currently unavailable. Depending on the positions of both JISC and the University of Nottingham, the source code may or may not be made available.

The following information is made available in case the source code is made available or for the information of future developers of the system.

The source code is within a Microsoft Visual C++ v6.0 workspace, structured to use the C++ object oriented class structure. The following main classes will briefly be described to give a vague idea of the structure of the project.

CSpaceApp & CMainFrame

These are the two main classes which handle the main window of the application. Coding controling views, menus and toolbars are within.

CSpaceDoc

This is the main document class which contains all the algorithm code and major functionality of the internals of the system.

CConstraint & CconstraintSet, CallocationsSet, Cresource & CresourceSet, Crooms & CroomsSet, Crequirements

These are all classes storing the database access to the file sets and there corresponding storage within the system.

Cgeneration, Cgene, Cindividual, CresourceGene

These are all classes representing parts of the algorithms storage of the space allocation solutions.

CfileSet

This class stores the information for the handling of filesets for the various input data groups.

CspaceWnd, CspaceView, CspaceListView, CspaceTreeView

These classes store the user interface information for the main window of the system.

Cgraph, CgraphTool, Cpicture

These classes handle the statistical graph coding.

CdetailsView, CtextView, CrequireSetView, CresourcesView, CroomsView, CconstraintSetView

These all represent different additional views that the system has available.

This description is a extremely brief and vague on, further information is stored within the source code in the form of comments.

5. GLOSSARY

Space	any area such as a room or building which can be allocated to a specific function or resource
Utilisation	a measure of how often a specific room is used for its allocated purpose. Low utilisation represents inefficient usage of space.
Constraint	a requirement or request expressed to the *system*, concerning the solution it produces. Constraints are specified and weighted by the *allocator* according to their importance.
System	the automated space allocation system. It's inputs are the *problem data* and *constraints* and its output is one or more specifications for the allocation of space.
Allocator	a person with expert knowledge of the space allocation process who can evaluate the desirability of any particular allocation and who can express to the *system* in terms of *constraints* what is required and desired of the output.
Problem Data	inflexible data describing *resources*
Resource	a staff member, room or any piece of equipment needed for consideration during the space allocation process.

6. REFERENCES

[1] EK Burke and DB Varley, *An Analysis of Space Allocation in British Universities*, Proceedings of PATAT '97, Toronto, Canada.

[2] EK Burke and DB Varley, *Automating Space Allocation in Higher Education*, Proceedings of SEAL '98, Canberra, Australia.