Preprint 11-071

EXPANDING THE LIMITATIONS OF THE MFIRE SIMULATION MODEL

S. Schafrik, Virginia Center for Coal and Energy Research, Blacksburg, VA
R. Ruckman, Mine Ventilation Services, Clovis, CA

ABSTRACT

In 2002 the authors updated the MFire model developed by the US Bureau of Mines. MineFire, developed by Mine Ventilation Services, Inc. (MVS), uses the MFire model to perform mine fire simulations on typical mine ventilation networks. The MineFire program is capable of producing visual schematics and other analysis tools for input and output of the MFire model. The MFire model, however, was limited to 500 branches and 10 fans. This paper discusses the newly modified MFire model that is now capable of modeling over 5000 branches and over 400 fans among other expansions and improvements. Given the long history and proven track record of MFire, the expansion did not modify the core calculation methods or results. This paper focuses on a general discussion of the MFire program and issues encountered in expansion of the algorithm. An example large network with simulation results is discussed.

INTRODUCTION

Fire and its resulting disturbances underground produce heat, fumes, and smoke which the ventilation system transports through the mine. Gases can be poisonous or explosive. Heat can change the intended flow of the ventilation system, transporting fumes along unexpected routes, or have other unintended consequences. The Recirculation of contaminated air is yet another consequence. Engineers need a tool that can

- Design, locate, and model components of the mine system, such as fire doors and fuel bays. Model fires in shops or fuel bays, or test gas dispersion (stench) for possible scenarios.
- Train, teach, and show mine personnel how a fire affects the mine. Indicate changes to the ventilation, time available for escape, and how fast the fume front moves.
- Investigate or estimate fires or possible fire scenarios. Predict what could happen or what has happened and analyze results.

Commercial development of mine fire simulation software has been difficult due to a limited market and relatively high development costs. In the 1980's the former United States Bureau of Mines commissioned a mine fire simulation software package. Michigan Technological University developed the package known as MFire. Although several modifications and updates were made to the program though 1995 (ending in version 2.20), neither MTU nor the USBM made efforts to provide a user interface that would facilitate and control data input and organize output data for direct visualization. Several non-US mining organizations have produced visually interactive mine ventilation/fire simulators; however, these programs are not currently available in the English language. Recently, there have been several research efforts to bring MFire into modern computing.

In late 2001, Mine Ventilation Services, Inc. (MVS) personnel obtained a copy of the MFire package from the National Institute of Occupational Safety and Health (NIOSH). NIOSH had the research product as a result of its taking over certain mine safety and health research responsibilities from the former USBM. The package contained both executable files and source code for the MFire program. This allowed MVS to review the code and its operation to assess the suitability of preparing a user interface that would make the program easier to use and create output that could be quickly interpreted. At that time MVS brought together a team including the authors, to investigate bringing MFire into a commercial software product.

A challenge in increasing the limitations of the MFire program was having the FORTRAN 77 written program compile and run on modern computer architectures. The program was written and designed for the i386 architecture and compiled using Microway FORTRAN (Laage, Greuer, and Pomroy 1995). In May of 2002, after FORTRAN structure modification, a successful set of MFire programs was generated using Compaq Visual FORTRAN. These executable programs became known as the MineFire Kernel in MVS's MineFire Product. MineFire serves as a bridge between the MVS VnetPC ventilation package and the MFire program. Visualizations that are familiar to VnetPC users are available for both the input and output of a fire simulation.

In 2009, MVS tasked the authors with modifying the MFire Kernel. The main goal of the modification was to make the MFire 2.20 program capable of working with the much larger networks that VnetPC can create, simulate and visualize. The core of the process was increasing MFire limitations from 500 to 4,000 branches, 350 to 3,500 junctions, and 10 to 400 fans. In this paper, a case study is presented and discussed as an example which offers an example of how the newly modified MFire can operate with the expanded models of VnetPC. Now, for the first time, a model done in VnetPC can be directly used by MFire using MineFire as the pre- and post- processor.

BACKGROUND

The program MFire was developed at the Michigan Technological University under a contract from the former United States Bureau of Mines (Laage, and Yang 1991) (Laage, Greuer, and Pomroy 1995). The final version of the MFire program is 2.20, which was last updated in 1995 (Zhou and Luo 2010). The final compiled version will operate in DOS and is not Windows compatible. It does not operate on other operating system platforms. As described above, MVS modified the MFire 2.20 program to run on Pentium class processors.

A new product, MineFire was developed to bridge the connection to other MVS products including VnetPC. MineFire allows the user to simulate fires, heat flow, contaminant flow, and/or natural ventilation in underground ventilation networks using the familiar VnetPC graphical and tabular interfaces. Results are displayed both symbolically and numerically on the schematic. This gives the user flexibility in utilizing the software package as a training tool or other specific needs. In addition to the functionality of VnetPC 2007, MineFire includes the following:

- Ability to import an existing ventilation network from VnetPC 2000, 2003 or 2007, or start a new model from MineFire. Import DXF files from AutoCAD or other CAD programs just like VnetPC.
- Fully interactive schematic that allows editing and data entry of ventilation parameters, rock property data, and temperatures without using table views. Table views allow the user to cut and paste data to and from Excel or other spreadsheet programs, useful for performing calculations and returning the answers to the program.
- Dynamic results view in the schematic that shows the progression of fume fronts and changes to the ventilation parameters over

adjustable time increments. See the progression of the fire, and problems it might be causing.

- Ability to vary and control calculation parameters using by means of Control Card functions, including time increments, calculation precisions, use of the fan curve, and more.
- Ability to change the properties of a branch between an ordinary airway, fire branch, or fan branch during the course of an event through the use of a time table. Increased control of the simulation, to approximate complex situations.
- Ability to display in Imperial or SI units, and offer a conversion utility.

Basic Description

The MFire program performs normal ventilation network planning calculations and dynamic transient state simulation of ventilation networks under a variety of conditions. MFire simulates a mine's ventilation system and its response to altered ventilation parameters, such as the development of new mine workings or changes in ventilation control structures; external influences, such as varying outside temperatures; and internal influences, such as fires.

MFire performs the transient state simulation of a mine ventilation system as a stepped series of short duration steady state calculations with output becoming the input for the next step. The heat transfer model is constructed on the basis of the energy balance of airflow to pre-calculate the time dependent air temperatures at different locations.

The MFire program assumes instantaneous and complete mixing under the conditions of fully developed turbulent flow. Thus, stratification is not considered and the products of combustion are evaluated as evenly distributed. Although the program tracks the average concentration of physical contaminants, the physical impacts to the ventilation system are based on the mechanical effects of the thermal inputs. The program does not account for the combustion chemistry; hence, the fire duration, heat output, and products of combustion are based on input from the user.

MineFire will read a VnetPC network or can create a network, confirm the network validity, and create the input for the MFire Kernel, system, which consists of 4 subprograms. The first is the batch file, which controls the execution of the three executable programs and handles output errors from these programs. The first program, mfire0.exe, initializes files needed by the subsequent programs. The second program, mfire1.exe, handles the network and temperature portions of the analyses. This program is also executed at the end of the time dependent portion to develop the steady state results. The third program, mfire2.exe, handles the non-steady state, or transient, portions of the problem. The output data are collated into a single output file, mfire.out.

The mfire.out file is read in and parsed by the MineFire program. It is capable of then displaying the network at each simulation time step. The program allows users to define various types of fires and simulate individual fires in different locations throughout a model. Once the program is executed, the results selected by the user may be graphically displayed on the schematic. The user may select which parameter will be displayed, such as the fumes or temperature. The program allows for data to be reviewed for time intervals per-set by the user. The time intervals allow a user to graphically display the spread of contaminates and temperature through the system after the start of the fire. Based on the simulation, users may adapt individual systems to adjust mitigation and emergency plans accordingly. The program allows users to modify systems to simulate mitigation plans such as closing doors after a set time. These types of systems may be used to interpret the effectiveness of mitigation plans and determine if personnel may be potentially exposed to contaminates. A fire simulation model may be utilized as an important part in the development of emergency and mitigation plans and training personnel, as well as simulated past events. While fire modeling is an important tool, the program requires significant input and interpretation of the results and is not suited for real time events but rather best utilized for training and planning.

Mine Fire Research

The MFire program is widely accepted as a standard for mine fire predictions (Cheng, Ueng, and Liu 2001). There have been several studies to validate its accuracy (Wala, Dziurzynski, Tracz, and Wooton 1995) (Smith and Thimons 2009). The MFire program has been used in tunneling applications as well, including subway planning (Miclea 1991).

There have been notable modifications to the MFire program after 1995. At the University of Nevada Reno, under the direction of Dr. Frederick Harris, Jr., Lingjiang Cheng (Cheng 2000) and Lu-Chun Liao (Liao 2000) completed MS Degrees in Computer Science by converting the first and second half of MFire to C++. The purpose of these translations was to provide a ventilation simulation for a Mine Virtual Reality project. Lihong Zhou completed a PhD in Mining Engineering at West Virginia University (Zhou 2009) by creating what is termed MFire 2.30. This project included the addition of (1) a time dependent fire and t-square fire (Zhou and Luo 2010), (2) the consideration and prediction of smoke roll back, (3) the incorporation of a moving fire source, and (4) a rewriting of MFire in Microsoft Visual C++ with database connections for input and output. Dr. Andrzej Wala at the University of Kentucky has developed an expert system approach to mine fire simulation in recent years. This simulator is described to be used for instructing engineers (Wala 1996). Dr. Wala was also the author of a study which offers valid results from MFire (Wala, Dziurzynski, Tracz, and Wooton 1995). NIOSH has also funded a project to develop what is termed as MFire 3.0., which, according to NIOSH. " will be structured as a class library that can be re-used to build new simulation programs to model mine ventilation and fires. It will have discrete event simulation capabilities that allow mine ventilation and fires to be tracked over time" (NIOSH 2010).

In all recent studies, projects have focused on the re-writing MFire. It is clear that the source code of MFire does not lend itself to direct translation. Such a study was undertaken by the authors in 2002, which looked at converting the difficult to understand FORTRAN code and into both ANSI C and Microsoft Visual Basic. In both cases, it was found that the development costs were prohibitively high for a commercial product. The calculations currently done by the MFire program are accepted by the industry and regulators. To do a complete re-write into another language that has the exact same calculations was cost prohibitive. It has been established that the MFire tool is capable of performing its duties as is, just not on modern computer platforms. and visual user interfaces are important. MVS has taken the approach of maintaining the FORTRAN code to keep the accepted calculation engine in place. This approach is also taken by other software companies that create engineering software. Two examples are Gaussian 09 (Gaussian 2010) and Advanced Resources International's COMET (Advanced Resources International 2010).

By integrating the MFire model with accepted ventilation prediction software (McPherson 1993), development costs can be minimized while keeping the same calculation engine in the original designs. It does not matter which computer language is used to calculate a value, it is the calculation action itself which is important.

LIMITATIONS AND EXPANSION

Ensuring a consistent result with the established MFire program was the paramount concern when expanding the limitations. First, the authors started with the FORTRAN compiler. In 2002, Compaq Visual FORTRAN was used to create the binaries. This product has since been discontinued and incorporated into Intel Visual Fortran. For the most part, the previous compilation command of MFire was directly applicable. It was found that by making some changes to the source code all special compilation directives could be eliminated. This elimination ultimately makes MFire cross-compatible. The current MFire source code can be built on 32 bit 64 bit Windows and Linux without further modification. Further, it is now possible to take advantage of the Intel Multiprocessor model, enabling MFire to use more than one processor. The production version of MineFire will only include the Windows compilations, but companion products can be created. During debugging, the authors found it convenient to also use PGI Visual Fortran. The PGI product is more tolerant of FORTRAN language constructs and was useful for consistency checking.

MFire 2.20 allows networks of up to 500 branches, 350 junctions, and 10 fans. VNetPC is capable of working with ventilation networks that are on the order of 5,000 branches and 400 fans. This increased number in MFire is controlled by several variables and constants inside the program. FORTRAN 77 is not capable of dynamically allocating memory for arrays. Dynamic allocations are done in FORTRAN by reading and writing data to binary data files, the mechanisms for which are built into the language. The 500 branches are primarily controlled by the statement “NMX=500” in the CMMN1.DAT file, which is one of several such parameters. NMX is a constant parameter that is used throughout the MFire programs for loops and array declaration. There are approximately 70 arrays that use the NMX parameter. All of these arrays are kept in memory during program execution. In addition, there are several arrays and data initializations that are multiples of the one NMX parameter.

As a result, the relationship between this one parameter and the amount of memory that is used by the program is not linear. It is not possible to simply make each constant parameter 10 times larger in order to increase the entire program’s capacity by 10 times . Several arrays are multidimensional, having more than just a linear relationship between the dimensions. For the MFire Kernel it was found that increasing the solve routine’s capacity by approximately 10 times increases the amount of memory needed by approximately 100 times. Editing the parameters is complicated by the memory structure and loop structures used in the solve routines. Without careful consideration of loop escapes, infinite loops might be encountered by the user during the solve method.

Data input is by means of a flat data file. The input of data to the MFire routine is through a text file. This file contains the information needed to execute the programs. The basic data requirement for MFire is described as:

- Control Card I: basic information regarding the number of branches, the number of fans, time increments, and certain reference values.
- Airway Cards: information about each of the branches in the network.
- Junction Cards: information about each of the junctions in the network.
- Junction Cards II: data relating to those junctions situated in the atmosphere. That is, those junctions air enters and exits the mine.
- Fan Cards I, II and III: data related to each fan, the fan characteristics, and treatment of the curve boundary regions.
- Contamination Cards: contamination and heat source data active at the beginning of the problem.
- Control Card II: this card allows the user to set the default values of certain variables.
- Average Value Card: this card allows the user to set average values of certain parameters for the heat flow analyses.
- Time Table for Conditions Changes: these cards allow the user to set up the scenario time table for analysis.

In the above listing the term "card" is used as the traditional term for a line of data in a text file. Some of the data or control cards are optional or only apply to the particular simulation. Comment cards are also permitted in the input file.

Commas or spaces are allowed file line delimiters and are treated interchangeably. Multiple blank spaces are treated as a single space. Integer and Real values are also treated interchangeably, for instance 5.0 and 5 are equivalent, and the FORTRAN language interpreter will modify the value on input as necessary. Comments are allowed in the input files. Comment lines are denoted by starting with a $ symbol. Blank lines are ignored.

The read in routine for input is extremely important to the MFire engine and it is also one of the more complicated sections of the program. The program is tolerant of input cards being out of order. It also does not require the user to input the number of cards to expect. This portion of the MFire 2.20 program required a majority of the edits to handle a much larger input model. In addition, several discrepancies between the MFire Manual and the program were located. An example is the MADJ variable, the MFire Manual states, “MADJ is the maximum number of iterations in the temperature section. If its input is less than 10 or larger than 80, the default value of 15 will be used.” However, the source code clearly is using a range of less than 5 and greater than 80 to a default value of 10. Such discrepancies are noted in MineFire, but not corrected.

Following the successful completion of the MFire run, the output data are collected into a single output file. This file is given the fixed name MFIRE.OUT. A typical run produces an output file that contains the following information:

- The basic model data
- Initial network calculations
- Airflow reversal information
- Fan performance characteristics and operating points
- Data for contaminant concentration and temperature
- Non-steady state output
- Temperature and concentration at the ends of the branches
- Fume front information
- Contaminant/heat source status

These outputs are easily overflowing former FORTRAN Write and Format Statements. The Write and Format statements are used by FORTRAN programmers to output data in columnar and human readable forms. Format statements specify the type of data (e.g. text, integer, real, scientific, etc.) and the number of printable characters that are to be used for the display. For instance, I3 specifies an integer value with 3 spaces thus the integer 10 will be displayed as “ 10”, but 1,000 will be displayed as “***”. The asterisks indicate an overflow condition, and mask the real information. There are over a thousand write statements in the MFire programs, each one was evaluated for overflow conditions and corrected.

The most useful tool created during the upgrading process was a memory dump utility. The utility is capable of reading the MFire program’s memory and writing it to text files that are named for the variables or the class of variables. Arrays and single values are fully outputted into separate files that can be easily compared using text differencing programs. Output for single variables, such as integers and strings are put into a single text file. The variable names and values are in colon separated format to allow the complete variable to be placed on a line. Arrays are outputted into a file named by the variable name. The array index and values are put on the same line and is colon separated. Multidimensional arrays are also outputted in the same manner with all indexes and a single value per line. This format was not chosen for storage efficiency, it was chosen for speed of text comparison.

Using the memory dump tool it is possible to take snapshots of the inner workings of MFire. It is coupled with a truncation tool that allows for memory comparisons between the MFire 2.20 and the expanded MFire kernels. Changes to the source code were made very systematically and singular changes were made. Then the test case was run through the debug compilations and not only was the output files compared, but the memory dumps at various points around each change. As the number of changes grew, the memory comparisons became the only reliable means of confirming that the calculations were progressing identically. As described above the Write and Format statements where highly edited, this ruled out using the direct output.

During development several test cases that utilize different portions of the MFire kernel were used. A single case was used as the base case. This case was chosen because it was the original base case used in 2002. The case involves a cooling station, and a fire in a 51 branch model. Specifics of the model and results are not germane to this study. It is important to note the differences. The MFire program that was generated in 2002 using Compaq Visual Fortran and the identical source code compiled using Intel Visual Fortran yield slightly different results. The differences appear to be explainable as rounding errors as they do not appear in the integer values. For this reason, only the Intel Visual Fortran compiled programs were compared to each other. The results did lead to a test of various Fortran compilers with identical source code and input files. It was found that Intel Visual Fortran, PGI Visual Fortran, and several other Fortran compilers did not yield identical results. These differences are generally rounded out in the output files, but can be significant enough to change the last significant digit.

It is noted in this case because the convergence of the solve method has an error tolerance of $5x10^{-5}$. It is entirely possible that convergence can be achieved in an executable created by one compiler at a different loop than another. These differences are also were encountered during development after expanding the number of fans limitation. For researchers or developers that are converting the program to another programing language similar phenomenon can be encountered. It is apparent when looking at the program's data as stored in memory, but can be difficult to notice otherwise. The most difficult compiler to use with the MFire program was the Linux GNU Fortran compiler. Nearly no variable declarations are made in the MFire 2.20 source code. All variables are typed by the data that is put into them. There are some real values that are stored as double precision; however, most are single precision. These interchanges between the two leads to a lot of the rounding errors discussed here and are confusing to most compilers. The variables in the expanded MFire kernel are all typed and consistent with the MFire 2.20 in memory and therefore in calculation.

Another FORTRAN caused phenomenon encountered was that of the integer divide. This is the case of having a variable containing and integer in the numerator or divisor. It was commonly used in FORTRAN programming to determine odd or even and is complements by the modulus operator. However, some compilers and other programming languages treat this type of division differently. For instance, 5 / 2.0 can yield 2, 3 or 2.5, all are legitimate answers under different integer divide regimes. There are several integer divides in MFire, most having to do with output but a few with program control. It is an esoteric consideration, but one that must be considered when performing a re-engineering of the source code.

Using the memory dump tool, it was possible to systematically change the MFire source code to increase the limitations. The driving limitations were the number of branches and number of fans capable to be simulated. In progressing to that goal several other limits were expanded, such as the maximum number of iterations and the maximum allowed node number. The output file was modified in a way to put out more data but not to disturb the post-processing capabilities of MineFire. The current MFire kernel used by MVS is capable of taking the maximum allowable ventilation model from VnetPC and do simulations without modification.

Case Study for Usage of Expanded Model

The newly expanded model is part of the MVS services and is sold as an MVS product. MVS is currently using the program to provide services to mining companies around the world. MVS engineers used the newly expanded limitations of the MineFire program to conduct fire analyses for a large metal mine. This simulation was conducted for a model with 3015 branches, 2347 junctions, and 102 fans. The ventilation system is designed with a central ramp system used for personnel and material transportation to the mine and working areas. In the ventilation design, the primary connection ramps between the central system and the levels were designed to be equipped with fire doors but will remain open during normal operations. Fresh air is supplied from the central ramp system to the levels.

Using the MineFire program, a light vehicle fire was simulated in the central ramp system. After approximately five minutes the fume contaminates spread through the ramp system to the access ramps and the active mining levels. Approximately thirty minutes after the fire started, the fumes may contaminate a significant portion of the mine where personnel may be present. The analyses illustrated the rapid spread of contaminates through the system if no mitigation or fire suppressant equipment is utilized to where personnel may be present.

For mitigation analyses, the fire doors located in the access ramps, where contaminates spread from the central ramp system to the levels, were simulated as closed after ten minutes. While the contaminates spread through the system rapidly and to the access ramps in less than five minutes, it is assumed that personnel will not immediately activate the emergency procedures to close the fire doors. After the fire doors are simulate closed, the spread of contaminates are reduced and where fumes are present in the system may begin to dissipate through the ventilation exhaust system. By closing the fire doors the fume contaminates were limited; therefore reducing the potential threat of exposure to personnel underground.

This simulation illustrated the potential impacts a fire in a primary access of a mine may affect the system if it is uncontrolled or no mitigation is implemented. Through the use of program, fire doors were simulated to determine the effectiveness of a mitigation strategy. With the doors closed, the program predicts fumes will still be present in the system; however, the isolation and dispersion of the contaminates through the system may occur. The mitigation simulation suggests the implementation of fire doors should be required and mine personnel should consider implementing automated doors which may be closed quickly in the event of an emergency.

CONCLUSIONS

The MFire computer model developed by the USBM in the 1980's into the 1990's is an impressive program. The program has been used in mining and various tunneling application to accurately predict the ventilation network's reaction to a fire. The program is also able to describe the flow of contaminates and other effects from the fire. MVS had previously modified the MFire program and integrated it with the existing VnetPC ventilation software. The resulting commercial product, MineFire, was previously limited to the capabilities of the MFire program as previously developed. This paper described how the MFire program has been modified to increase program limitations from 500 to 4,000 branches, 350 to 3,500 junctions, and 10 to 400 fans. Many other improvements were made to the program based on the years of experience with the previous MFire. The resulting MFire kernel is included in the updated MineFire product. This program now more closely integrates with VnetPC, meaning a ventilation network can be simulated directly, without simplification or truncation. This approach varies from other approaches taken because it results in a commercial software application and was not a re-engineering of accepted simulation code and methodology.

REFERENCES

1. Advanced Resources International, http://www.adv-res.com/comet3-nonconventional-resources.asp Accessed Oct. 2010

2. Chang, X., L.W. Laage, and R.E. Greuer, 1990, A User's Manual for MFIRE: A Computer Simulation Program for Mine Ventilation and Fire Modeling, USBM, IC 9245, 171 pp.

3. Cheng, Lingjiang, 2000, Large Scale Software Transitions: A Case Study of the First Half of MFire, Master of Science in Computer Science Thesis, University of Nevada Reno

4. Cheng, L.H., Ueng, T.H., Liu, C.W., 2001. "Simulation of Ventilation and Fire In The Underground Facilities". Fire Safety Journal 36 (6), 597–619.

5. Gaussian, http://www.gaussian.com/ Accessed Oct. 2010

6. Laage, L. W. and H. Yang, 1991, "Mine Fire Experiments at the Waldo Mine: Heat Flow," Chapter 7, *Proceedings of the 5th US Mine Ventilation Symposium,* June 3-5, 1991, Morgantown, WV, Society for Mining, Metallurgy, and Exploration, Inc.

7. Laage, L.W., R.E. Greuer, and W.H. Pomroy, 1995, MFIRE Users Mannual Version 2, U.S.Bureau of Mines Twin Cities Research Center, Minneapolis, MN, 112 pp.

8. Liao, Lu-Chun, 2000, Large Scale Software Transitions: A Case Study of the Second Half of MFire, Master of Science in Computer Science Thesis, University of Nevada Reno

9. McPherson, M.J., 1985, "The Analysis and Simulation of Heat Flow into Underground Airways," International Journal of Mining and Geological Engineering, 4, 165-196

10. McPherson, M.J., 1993, Subsurface Ventilation and Environmental Engineering, Chapman & Hall, New York, 905 pp.

11. Miclea, P.C. Application of SES and MFIRE computer simulation programs for tunnel fire modeling—a comparative study, ICF Kaiser Engineers Inc., Oakland, CA, USA (1991)

12. NIOSH, http://www.cdc.gov/niosh/mining/mineract/contracts/200-2009-30794.htm, Accessed Oct. 2010

13. Shepherd, W.C.F., S. Jones, 1952, "Investigations into Underground Fires", Proc. of 7th International Conference of Directors of Safety in Mine Research, pp.1-40.

14. Smith, Alex, Thimons, Edward, 2009, "A Summary of U.S. Mine Fire Research" 2010 SME Annual Meeting and Exhibit, February 28 - March 3, Phoenix, Arizona.

15. Wala, A.W., M. Dziurzynski, J. Tracz, and D. Wooton, 1995, "Validation Study of the Mine Fire Simulation Model," Proc. of 7th US Mine Ventilation Symposium, A. Wala (ed.), pp.199-206

16. Wala, Andrzej, 1996, "Teaching The Principles Of Mine Fire Using An Intelligent Computer-Aided Instruction",Proceedings of Frontiers in Education Conference, 1996. FIE '96. 26th Annual Conference. Salt Lake City, Utah

17. Zhou, L., Luo, Y., 2010, "Application of the t-squared fire model in MFIRE," Proc. of 13th United States/North American Mine Ventilation Symposium Hardcastle & McKinnon (Eds.)

18. Zhou, Lihong. 2009, Improvements of the Mine Fire Simulation Program MFire, Doctor of Philosophy in Mining Engineering Dissertation.