

РАСПАРАЛЛЕЛИВАНИЕ И ОПТИМИЗАЦИЯ ВЫПОЛНЕНИЯ РАСЧЕТОВ ПРОЦЕССА РАЗВИТИЯ ПОЖАРА НА ОСНОВЕ ТРЕХМЕРНЫХ КЛЕТОЧНЫХ АВТОМАТОВ

© 2014

В.Н. Рудницкий, доктор технических наук, профессор, заведующий кафедрой системного программирования
Черкасский государственный технологический университет, Черкассы (Украина)

Е.А. Мельникова, кандидат физико-математических наук,
доцент кафедры прикладной математики и информатики

Тольяттинский государственный университет, Тольятти (Россия)

М.О. Пустовит, научный сотрудник

Академия пожарной безопасности имени Героев Чернобыля ГСЧС Украины, Черкассы (Украина)

Ключевые слова: распараллеливание; оптимизация алгоритмов; клеточные автоматы; моделирование; развитие пожара.

Аннотация: В работе представлена модель процесса развития пожара на основе трехмерных клеточных автоматов. Предложены способы распараллеливания и оптимизации алгоритмов для реализации моделей распространения огня и дыма, позволяющие повысить скорость получения результатов моделирования для дальнейшего использования в компьютеризированном тренажере подготовки пожарного.

Математические модели распространения пожара и задымления позволяют имитировать реальное состояние объекта с высокой точностью в реальном времени, создавать практически все возможные ситуации, в том числе вводить возможные аварийные ситуации и режимы работы оборудования для отработки действий персонала в особых режимах и ситуациях, создавать визуальную картину окружающего пространства и его изменения в процессе работы [1].

Одним из методов моделирования распространения пожара и задымления является метод клеточных автоматов (КА). Основными преимуществами метода является высокая скорость обработки данных, что обеспечивается математической моделью, достаточная достоверность полученных результатов, отображение результатов в графическом виде.

Авторы [2] используют КА для расчета распространения пожара и задымления в больших зданиях. Основным недостатком данного метода является значительный размер клетки, и как следствие – низкая точность прогнозирования. Более эффективный подход на основе применения модели распространения пожара с помощью метода клеточных автоматов предлагают авторы [3, 4]. Положительной чертой является возможность масштабирования клетки для получения более точных результатов.

Целью исследования является оптимизация алгоритмов разработанной модели развития пожара методом распараллеливания для использования в компьютеризированном тренажере подготовки пожарного спасателя.

В разработанной модели объем этажа здания представляется в виде множества элементов – расчетных единиц объема (соответствующих кубическому метру, сантиметру и т.п.), каждый из которых имеет свои характеристики горючей нагрузки, в том числе, линейную скорость распространения пламени. Распространение пожара в том или ином направлении от источника воспламенения определяется вероятностью возгорания каждого такого элемента. Таким образом, разработанная модель является имитационной вероятностной моделью [4].

Пространственная область процесса распространения пожара (полигон) представляет собой трехмерное пространство, которое представлено трехмерным массивом со следующими максимальными значениями по каждому измерению: x_{max} – число индексов по измерению «x»; y_{max} – число индексов по измерению «y»; z_{max} – число индексов по измерению «z» [5]. Принадлежность индекса к соответствующему измерению определяется следующим образом:

$$\begin{aligned} x &\in \{1, 2, \dots, x_{max}\}; \\ y &\in \{1, 2, \dots, y_{max}\}; \\ z &\in \{1, 2, \dots, z_{max}\} \end{aligned}$$

Каждый компонент трехмерного массива представляет собой клеточный автомат, моделирующий процесс распространения пожара (задымления). Конфигурация расположения клеточного автомата с ортогональными с ним клеточными автоматами определяется его координатами в трехмерном векторном пространстве. Существует ограниченное количество конфигураций $K = \{k_1, k_2, \dots, k_{26}\}$

$$k|_{k \in K} = \begin{cases} k_1(0 < x_0 + 1 < x_{max}) \wedge (z = z_0) \wedge (y = y_0); \\ k_2(0 < x_0 - 1 < x_{max}) \wedge (z = z_0) \wedge (y = y_0); \\ k_3(0 < y_0 + 1 < y_{max}) \wedge (x = x_0) \wedge (z = z_0); \\ k_4(0 < y_0 - 1 < y_{max}) \wedge (x = x_0) \wedge (z = z_0); \\ k_5(0 < z_0 + 1 < z_{max}) \wedge (x = x_0) \wedge (y = y_0); \\ k_6(0 < z_0 - 1 < z_{max}) \wedge (x = x_0) \wedge (y = y_0). \end{cases} \quad (1)$$

где x_0, y_0, z_0 – координаты источника воспламенения.

Общая конфигурация соседства клеточного автомата показана на рис. 1 (а, б, в). Конкретная конфигурация определяется значением координаты клеточного автомата, который находится в невидимой зоне на рис. 1а и зависимости (1).

Координаты клеточных автоматов, расположенных в углах полигона (рис 1б) определяются зависимостью (2).

$$k|_{k \in K} = \begin{cases} k_7(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_8(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_9(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{10}(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{11}(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{12}(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{13}(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{14}(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}) \end{cases} \quad (2)$$

Взаимное расположение клеточного автомата с соседними КА, имеющими координаты, определенные в (2), показано на рис 1б.

Координаты клеточных автоматов, расположенных на ребрах полигона и удаленных от его углов, определяются как:

$$k_{k \in K} = \begin{cases} k_{15}(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (z = 0); \\ k_{16}(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 + 1 < y_{max}) \wedge (z = 0); \\ k_{17}(0 < x_0 + 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (z = 0); \\ k_{18}(0 < x_0 - 1 < x_{max}) \wedge (0 < y_0 - 1 < y_{max}) \wedge (z = 0); \\ k_{19}(x = 0) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{20}(x = 0) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{21}(x = 0) \wedge (0 < y_0 + 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{22}(x = 0) \wedge (0 < y_0 - 1 < y_{max}) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{23}(0 < x_0 + 1 < x_{max}) \wedge (y = 0) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{24}(0 < x_0 - 1 < x_{max}) \wedge (y = 0) \wedge (0 < z_0 + 1 < z_{max}); \\ k_{25}(0 < x_0 + 1 < x_{max}) \wedge (y = 0) \wedge (0 < z_0 - 1 < z_{max}); \\ k_{26}(0 < x_0 - 1 < x_{max}) \wedge (y = 0) \wedge (0 < z_0 - 1 < z_{max}). \end{cases} \quad (3)$$

Взаимное расположение клеточного автомата с соседними КА, имеющими координаты, определенные в (3), показано на рис. 1в.

Вероятности воспламенения определяется исходя из горящих материалов, расположенных в соседних клетках. Эти вероятности хранятся в многомерном массиве для всей сетки: три индекса (x, y и z) указывают положение в сетке относительно исходной ячейки (x₀, y₀, z₀), а четвертый k показывает положение соседних клеток.

$$FIXED_PROB_F(1:N, 1:M, 1:L, 1:26)$$

где элементы являются вероятностями

$$FIXED_PROB_F(x, y, z, k) \in [0,1],$$

N, M и L это размеры сетки (наша сетка имеет NxMxL клеток), у каждой клетки имеются 26 соседей, которые являются равноценными (рис. 1).

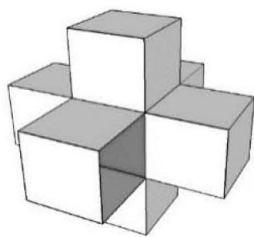


Рис. 1а. Обобщенная конфигурация соседства клеточного автомата с координатами КА, определенным в (1)

Перед перерисовкой каждого кадра анимации по формуле (4) вычисляются вероятности возгорания каждого элемента с учетом его характеристик и наличия соседних горящих элементов:

$$FIXED_PROB_F = \frac{V \cdot FP}{4} \quad (4)$$

где V – линейная скорость распространения пламени, выраженная в единицах (элемент/кадр анимации); FP – параметр, характеризующий количество и относительное расположение соседних горящих элементов. Он может принимать значения в интервале [0...50]

$$FP = 3n_o + 2n_d + n_a,$$

где n_o – количество горящих элементов, расположенных ортогонально по отношению к рассматриваемому, n_d – количество горящих элементов, расположенных по диагонали относительно рассматриваемого элемента, n_a – количество горящих элементов, расположенных в углах рассматриваемого элемента.

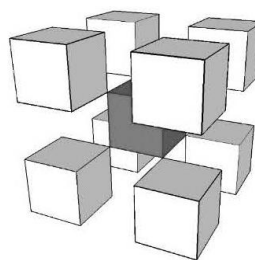


Рис. 1б. Обобщенная конфигурация соседства клеточного автомата с координатами КА, определенным в (2)

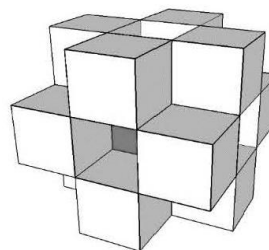


Рис. 1в. Обобщенная конфигурация соседства клеточного автомата с координатами КА, определенным в (3)

Переходные правила относительно КА распространения пожара для ячейки (x, y, z) на шаг по времени t, обозначается A(x, y, z)_t (5):

$$A_{(x,y,z)_t} = \begin{cases} 0 \text{ if } (A(x, y, z)_{t-1} = 0) \\ \text{or } (A(x, y, z)_{t-1} = 2) \text{ and } (time_expired(x, y, z) = 0) \\ 1 \text{ if } (A(x, y, z)_{t-1} = 1) \\ 2 \text{ if } (A(x, y, z)_{t-1} = 1) \text{ and } (random < FIXED_PROB_F) \\ \text{or } (A(x, y, z)_{t-1} = 2) \text{ and } (time_expired(x, y, z) > 0) \end{cases} \quad (5)$$

где time_expired (x, y, z) является счетчиком, который уменьшается от максимального значения τ_{max} на каждый временной шаг, пока значение КА не достигнет 0.

Клеточный автомат распространения дыма в трехмерном пространстве аналогичен автомату, разработанному для распространения огня с некоторыми корректировками в параметрах.

Как и распространение огня, распространение дыма характеризуется параметром матрицы, аналогичным $FIXED_PROB_F(x, y, z, k)$ и обозначается $FIXED_PROB_S(x, y, z, k)$.

Учитывая, что продукты горения поднимаются вверх, мы можем сделать вывод, что накопление дыма в вертикальном измерении происходит постепенно. Вследствие этого клеточный автомат распространения дыма в трехмерном пространстве приобретает те же параметры, что и КА распространения дыма в двумерном пространстве с постепенным добавлением значений по оси z .

Мы принимаем, что дым остается на неопределенный период времени в конкретной клетке (т.е. мы считаем, что дым присутствует в клетке, до тех пор, пока пожар не погашен). Основываясь на этих предположениях, переходные правила в отношении автомата дыма следующие:

$$B(x, y, z)_t = \begin{cases} 0 & \text{if } (B(x, y, z)_{t-1} = 0) \text{ or } (A(x, y, z)_{t-1} = 0) \\ 1 & \text{if } (B(x, y, z)_{t-1} = 1) \\ 2 & \text{if } (B(x, y, z)_{t-1} = 1) \text{ and } (\text{random} < FIXED_PROB_S) \\ & \text{or } (B(x, y, z)_{t-1} = 2) \text{ or } (A(x, y, z)_{t-1} = 2) \end{cases}$$

Перед перерисовкой каждого кадра анимации по формуле (6) вычисляются вероятности возгорания каждого элемента с учетом его характеристик и наличия соседних горящих элементов

$$FIXED_PROB_S = \frac{V \cdot SP}{4} \quad (6)$$

где V – линейная скорость распространения дыма, выраженная в единицах [элемент/кадр анимации]; SP – параметр, характеризующий количество и относительное расположение соседних элементов. Он может принимать значения в интервале $[0..26]$, поскольку «удельный вес» задымленных элементов принимается одинаковым. $SP=n$, где n – количество задымленных элементов.

Существует большое количество статистических алгоритмов, пригодных для эффективного распараллеливания и повышения производительности программного кода. Достаточно часто такие алгоритмы работают совместно с набором псевдослучайных чисел, например, при случайной выборке каких-то значений из большого массива данных.

В программном коде многократно повторяются вычисления, итерации которых зависят от различных наборов случайных чисел. Такой код очень хорошо подходит для распараллеливания. В целом, если каждая итерация будет выполняться на отдельном ядре, время выполнения кода существенно снизится.

Качественно сгенерировать массив случайных чисел с помощью нескольких параллельных потоков можно различными способами. Можно воспользоваться готовыми потокобезопасными параллельными функциями, например, библиотекой масштабируемых параллельных генераторов случайных чисел ($SPRNG$), о которой рассказано в статье [6]. Ещё одним вариантом является

запуск нескольких копий исходной функции генерации чисел. Метод, обсуждаемый ниже, обеспечивает генерацию качественного набора случайных чисел. Кроме того, в результате его использования создаётся один и тот же набор случайных чисел, независимо от количества одновременно запущенных функций – будь то одна функция в последовательном коде или несколько функций в параллельном коде:

Ниже приведён разработанный нами параллельный алгоритм реализации модели распространения огня и дыма в трехмерном пространстве (рис. 2).

К сожалению, даже параллельный алгоритм не позволяет реализовать эту модель в режиме реального времени на средних в аппаратном плане компьютерах, в связи с очень большим количеством элементов, состояние которых необходимо анализировать (64000 в каждом кубическом метре объема здания). При моделировании пожара даже в небольшом офисе по этому алгоритму на современных компьютерах удастся достичь скорости всего 10–15 кадров в секунду.

Кроме того, при таком обращении к элементам при выполнении условий (5, 6) и возгорании элемента автоматически увеличивается значение FP и SP для соседних элементов, и вероятность изменения их состояния на текущем шаге вычисляется с уже измененным его значением. Так при $V=V_{max}$ вся площадь может быть охвачена огнем всего за один шаг счета.

Обе указанные проблемы удалось решить путем оптимизации алгоритма за счет использования массивов особого типа – списков [6].

Создаются четыре объекта этого класса:

CheckList – список указателей на элементы, которые еще не горят (или не задымлены) и расположены вплотную к фронту пожара (загорание которых собственно имитирует распространение пламени в том или ином направлении);

NewList – список указателей на элементы, которые предстоит зажечь на этом шаге счета, поскольку выполняется условие (5, 6);

FireList – список указателей на элементы (их совокупность является объемом пожара), которые уже горят.

SmokeList – список указателей на элементы (их совокупность является объемом задымления), которые задымлены.

Таким образом, необходимые операции делаются не со всеми элементами всей площади здания, а только с теми, которые непосредственно вовлечены в модель и являются объемом пожара или задымления.

Реализация модели происходит по следующему алгоритму:

– обрабатывается список *CheckList*: вычисляется вероятность возгорания (задымления) каждого элемента, при выполнении условий (5,6) элемент переносится в список *NewList*;

– обрабатывается список *NewList*: элементы списка помечаются красным (серым) цветом, во всех соседних элементах увеличивается значение FP (SP), и они заносятся в список *CheckList*;

– все элементы из *NewList* переносятся в *FireList* (*SmokeList*)

– обрабатывается список *FireList* (*SmokeList*): вычисляется площадь пожара и другие параметры,

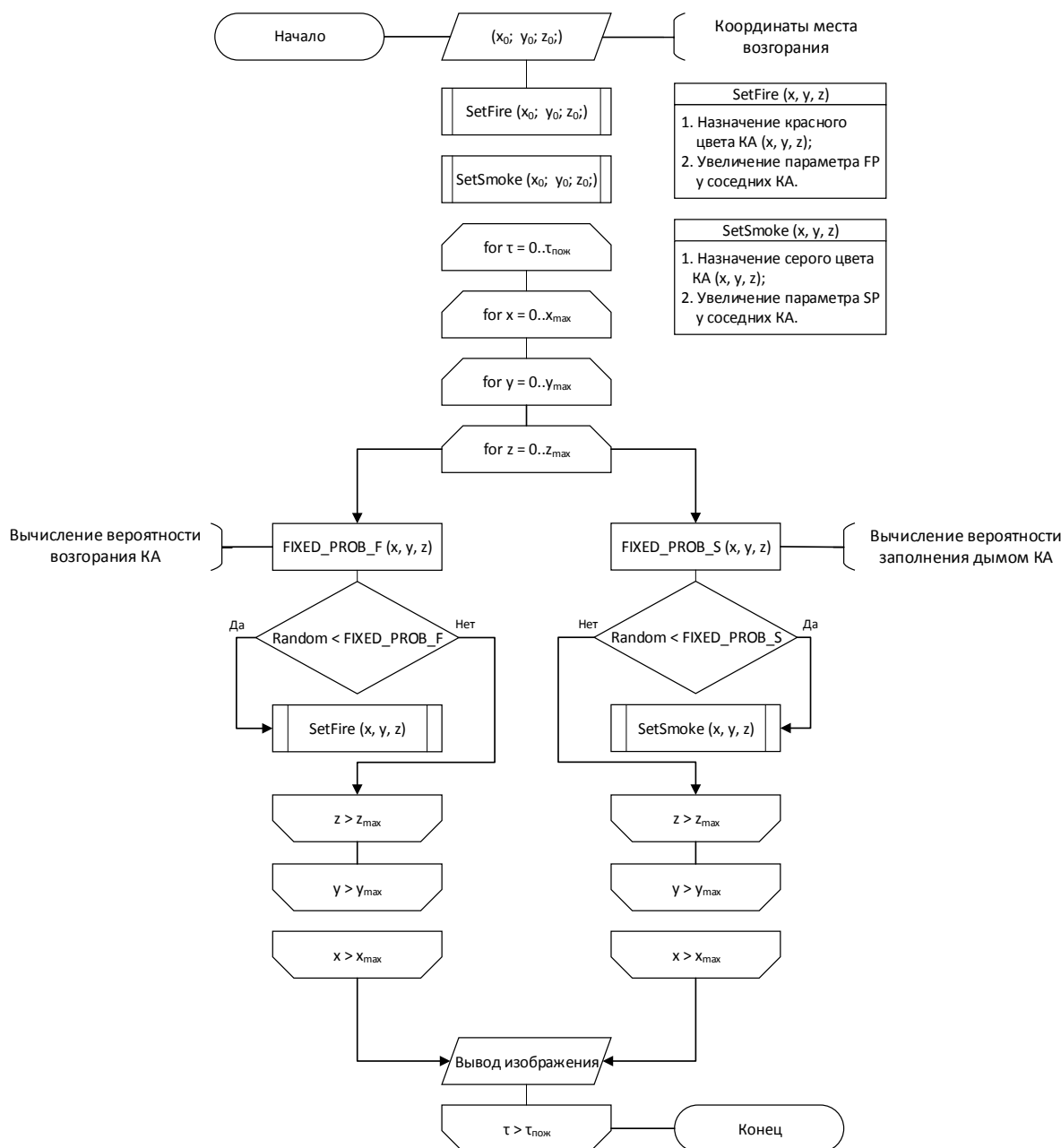


Рис. 2. Параллелизованный алгоритм модели распространения огня и дыма в трехмерном пространстве

необходимые для реализации интегральной модели; пропорционально скорости выгорания уменьшается масса горючего материала, при выгорании всей массы, приходящейся на эту расчетную единицу площади, она удаляется из списка (затухание).

Таким образом, благодаря возможности присваивать каждому элементу свои характеристики, можно имитировать распространение пожара при одновременном нахождении в помещении любого количества различных видов горючего, в том числе возгорание дверей и перегородок с последующим переходом пожара в соседнее помещение.

Подводя итог вышеизложенному, можно сказать, что полученная вероятностная модель распространения

пожара в трехмерном пространстве, позволяет получить площадь и объем (количественно) и форму пожара при любых введенных пользователем конфигурациях помещений, характеристиках горючего и произвольном количестве источников зажигания.

Благодаря методам распараллеливания и оптимизации алгоритмов удалось получить те значения скоростей просчёта модели на разных конфигурациях компьютеров, которые разрешают работать с результатами моделирования в режиме реального или ускоренного времени. Это в свою очередь даёт возможность использовать полученные в статье результаты при разработке компьютеризированных комплексов подготовки пожарного.

Работа частично поддержана ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 годы (соглашение №14.В37.21.1934).

СПИСОК ЛИТЕРАТУРЫ

1. Ю.А. Кошмаров Прогнозирование опасных факторов пожара в помещении: учебное пособие. – М: Академия ГПС МЧС России, 2000. с. 118.
2. Daniel-Ioan Curiac Fire Spreading Simulation in Large Buildings Based on Cellular Automata // *Advances In Dynamical Systems And Control*, vol. 45, no 3, pp. 159–162, 2010.
3. Мельников Б.Ф., Пивнева С.В. Математическое моделирование принятия решений в различных предметных областях // *Вектор науки Тольяттинского государственного университета*, 2010, № 3, с. 22–24.
4. Соколов С.В. Методологические основы разработки и использования компьютерных имитационных систем для исследования деятельности и проектирования аварийно-спасательных служб в городах [Текст]: дис. док. техн. наук: 05.13.10 / Соколов С.В. – М., 1999, 295 с.
5. Филиппенко И.Г., Головки А.В. Клеточные автоматы – основа построения математической модели процесса распространения пожара // *Восточно-Европейский журнал передовых технологий*, 2010, Вып. № 3/5 (45), с. 8–13.
6. Жабин В.И. Метод распараллеливания процессов в вычислительных системах // *Вестник Национального технического университета Украины «Киевский политехнический институт»*. Информатика, управление и вычислительная техника, 2000, № 34, с. 136–142.

PARALLELIZATION AND OPTIMIZATION OF FIRE PROPAGATION PROCESS CALCULATION BASIS ON THREE-DIMENSIONAL CELLULAR AUTOMATA

© 2014

V.N. Rudnitskiy, doctor of technical sciences, professor, head of the chair «System Programming»

Cherkassy State Technological University, Cherkassy (Ukraine)

E.A. Melnikova, candidate of physico-mathematical sciences,
associate professor of the chair «Applied Mathematics and Informatics»

Togliatti State University, Togliatti (Russia)

M.O. Pustovit, research associate

Academy of Fire Safety named after Chernobyl Heroes, Cherkassy (Ukraine)

Keywords: parallelization; optimization algorithms; cellular automata; modelling; fire propagation.

Annotation: The paper presents a model of development based fire three-dimensional cellular automata. Parallelization and optimization of algorithms for implementing models spread of fire and smoke offered. These methods allow to increase the production rate of the simulation results for further use in a computerized simulator training for firefighter.