УДК 004.93

А.Д. Дрюк[1], Е.И. Кучеренко[2]

[1] ХНУРЭ, г. Харьков, Украина, sanya40@ukr.net

[2] ХНУРЭ, г. Харьков, Украина, тел. 7021337

СТОХАСТИЧЕСКИЕ МОДЕЛИ И МЕТОДЫ СУБОПТИМАЛЬНОЙ МАРШРУТИЗАЦИИ СЛОЖНЫХ ОБЪЕКТОВ

Разработана стохастическая модель перемещения группы сложных объектов по территории производственного помещения. Сформулирована задача перемещения грузов сложными объектами за фиксированное время. Предложен метод, который за полиномиальное время находит субоптимальные траектории объектов, позволяющие решить данную задачу с вероятностью не менее пороговой. Приведена UML-диаграмма классов приложения, реализующего предложенный метод.

МОБИЛЬНЫЙ ОБЪЕКТ, МАРШРУТИЗАЦИЯ, РАЗБИЕНИЕ НА ЗАДАНИЯ, СУБОПТИМАЛЬНАЯ ТРАЕКТОРИЯ, АСИМПТОТИЧЕСКАЯ СЛОЖНОСТЬ

Введение

Проблемы построения систем управления едва ли не самые трудные при построении гибких производственных систем (ГПС). Однако практика создания, освоения и эксплуатация сначала простых, а затем более сложных систем в промышленности, а также опыт теоретических исследований специфических для ГПС проблем управления позволят создать необходимую теоретическую и инструментальную базу для быстрого последующего развития инженерной практики в области комплексной автоматизации производства на основе использования наиболее прогрессивных форм его организации, применения самой передовой техники и технологий [1].

Одной из подсистем системы управления ГПС является система управления транспортным и погрузочно-разгрузочным процессом. Этот процесс входит во все фазы производства, так как связан с распределением предметов труда и без принципиально новой транспортной техники и эффективной системы транспортирования, реализующей новую транспортную технологию, невозможна высокоразвитая промышленность.

Наиболее эффективным средством транспортировки грузов являются сложные объекты, к которым, в первую очередь, следует отнести мобильные объекты. В работе [2] описан один из методов оптимизации перемещения группы мобильных объектов, основанный на использовании нейронных сетей. Однако время работы этого метода очень сложно оценить, поскольку время обучения нейронной сети существенно зависит от ее входных параметров.

В данной работе предлагается другой подход, основанный на стохастической модели перемещения мобильных объектов по территории производственного помещения с заданной инфраструктурой. Метод оптимизации маршрутов мобильных объектов, предлагаемый авторами работы, позволяет находить субоптимальное решение поставленной задачи за полиномиальное время.

Одним из преимуществ автоматизации транспортных средств является сокращение времени ожидания грузов на рабочих местах и сокращение затраты энергии за счет эффективного планирования маршрутов движения. Поэтому в системе управления мобильными транспортными объектами одно из основных мест занимает планирование их перемещения, что является важным и актуальным.

Автоматизация процесса планирования перемещения, при минимизации затрат времени на подготовительно-заключительные операции и ускорении процесса переключения транспортного мобильного объекта с одного производственного задания на другое, является основой для организации гибкого производства [3].

Целью данной работы является оптимизация управления мобильными объектами на производстве, что позволит сократить потребление энергии и увеличить срок эксплуатации мобильных объектов.

1. Постановка задачи исследования

Пусть территория производственного помещения представляет собой прямоугольное поле, разбитое на единичные клетки. На этой территории находится L грузов и K мобильных объектов, способных погружать, разгружать и перемещать грузы. Пусть $G=\{g_1,g_2,...,g_L\}$ – множество грузов, а $R=\{r_1,r_2,...,r_K\}$ – множество мобильных объектов.

В связи с завозом новой партии грузов требуется переместить некоторое подмножество грузов $G_m \subseteq G$ в другие позиции, используя мобильные объекты из множества R. При этом все время выполнения этого задания не должно превышать предельного времени τ, которое осталось до прибытия новых грузов. Если же переместить все грузы за время τ невозможно, необходимо модифицировать задачу, чтобы задание было выполнено.

2. Разработка подходов к перемещению мобильных объектов по территории производства

Пусть территория производственного помещения представляет собой прямоугольное поле

размером $N \times M$, где $N = 4k + 2$ и $M = 3l + 1$ $(k, l \in \boldsymbol{N})$.

Территория разбита на единичные клетки. В некоторых из этих клеток находятся грузы и мобильные объекты, которые могут перемещать грузы. При этом в одной клетке может находиться не более одного груза или мобильного объекта.

Определим координату клетки как пару натуральных чисел (x, y), где x – номер клетки по счету сверху, а y – номер клетки по счету слева. Так, верхняя левая клетка имеет координаты $(1,1)$, а правая нижняя – координаты (N, M).

Пусть S – множество клеток территории помещения. Разобьем множество S на 2 класса – *дороги* D и *хранилища* H – так, что:

$$H = \{(x, y) \in S : ((x = 3k - 1) \vee (x = 3k)) \wedge \wedge ((y = 4l - 1) \vee (y = 4l)) | k, l \in \boldsymbol{N}\}, \quad (1)$$

$$D = \{(x, y) \in S : (x = 3k - 2) \wedge \wedge ((y = 4l - 3) \vee (y = 4l - 2)) | k, l \in \boldsymbol{N}\}. \quad (2)$$

Очевидно, что $H \cap D = \varnothing$ и $H \cup D = S$.

На рис. 1 приведена схема территории производственного помещения при значениях $k = 2$ и $l = 3$ $(N = M = 10)$, где серым цветом обозначены клетки хранилищ.

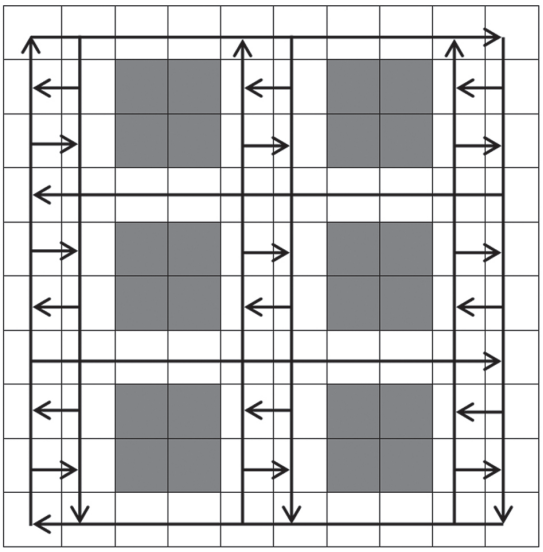

Рис. 1. Схема территории производственного помещения при $N = M = 10$

Пусть выполняются следующие правила:

1) Груз, не перемещаемый мобильным объектом, может находиться только в хранилище.

2) Мобильный объект может находиться в клетке хранилища в том и только том случае, когда он забирает груз, расположенный в этой клетке. Во всех остальных случаях грузы перемещаются только по дорогам.

3) Мобильные объекты могут перемещаться по дорогам только в определенных направлениях. Пусть (x, y) – текущая координата объекта. Тогда:

– если $y = 4k - 3$, где $k \in \boldsymbol{N}$, то объекту запрещается двигаться вниз. Если при этом x нечетно, то объекту также запрещено двигаться вправо;

– если $y = 4k - 2$, где $k \in \boldsymbol{N}$, то объекту запрещается двигаться вверх. Если при этом x четно, то объекту также запрещено двигаться влево;

– если $x = 6l - 5$, где $l \in \boldsymbol{N}$, то объекту запрещается двигаться влево;

– если $x = 6l - 2$, где $l \in \boldsymbol{N}$, то объекту запрещается двигаться вправо.

Кроме того, мобильным объектам запрещается покидать территорию помещения и въезжать на территорию хранилища без погрузки.

На рис. 1 стрелочками показаны разрешенные направления движения мобильных объектов по дорогам.

4) Все объекты движутся с одинаковой скоростью независимо от того, перемещают они груз или нет. Время, за которое объект достигает соседней клетки, обозначим через t_0.

5) В случае, если два объекта одновременно хотят попасть в одну и ту же клетку, приоритет имеет тот, который движется в вертикальном направлении (вверх или вниз).

6) Время погрузки и разгрузки всех объектов одинаково и равно t_l, причем t_l делится нацело на t_0.

7) В каждый момент времени мобильный объект может перемещать не более одного груза.

8) Время от времени объекты ломаются. Если мобильный объект сломался, он не может двигаться и должен ждать на одном месте, пока его отремонтируют. Время ремонта всех объектов одинаково и равно t_r, причем t_r делится нацело на t_0.

9) Если мобильный объект сломался между двумя клетками, перед началом ремонта его передвигают в центр ближайшей клетки. В таком случае будем говорить, что объект сломался *в окрестности* этой клетки. При этом время передвижения сломанного объекта включено во время его ремонта.

10) Расстояние, которое проходит объект между двумя последовательными поломками, – случайная величина ω, имеющая *экспоненциальное распределение* с параметром λ, одинаковым для всех объектов: $\omega \sim Exp(\lambda)$. Предполагаем также, что изначально все объекты отремонтированы.

11) В начальный момент времени все объекты отремонтированы и после ремонта еще не передвигались.

12) Вероятность $P(\omega \le t_0)$ того, что интервал между двумя поломками не превысит t_0, пренебрежимо мала.

3. Разработка модели

Пусть на территории помещения находится K мобильных объектов $r_1, r_2, \ldots, r_K$ и L грузов $g_1, g_2, \ldots, g_L$. Обозначим начальные координаты

i -го объекта через $\left(x_{r_i}^{(0)}, y_{r_i}^{(0)}\right)$, а начальные координаты j -го груза через $\left(x_{g_j}^{(0)}, y_{g_j}^{(0)}\right)$. Будем предполагать, что начальное расположение мобильных объектов и грузов удовлетворяет правилам:

$$\left(x_{r_i}^{(0)}, y_{r_i}^{(0)}\right) \in D \quad \forall i = \overline{1,K}; \tag{3}$$

$$\left(x_{g_j}^{(0)}, y_{g_j}^{(0)}\right) \in H \quad \forall j = \overline{1,L}. \tag{4}$$

Разобьем множество объектов $R = \{r_1, r_2, ..., r_K\}$ и множество грузов $G = \{g_1, g_2, ..., g_L\}$ на T непересекающихся классов:

$$R = R_1 \cup R_2 \cup \ldots \cup R_T;\ R_i \cap R_j = \varnothing\ \forall i, j : i \neq j, \tag{5}$$

$$G = G_1 \cup G_2 \cup \ldots \cup G_T;\ G_i \cap G_j = \varnothing\ \forall i, j : i \neq j. \tag{6}$$

Будем считать, что мобильный объект r_i, принадлежащий классу R_q, может перемещать груз g_j тогда и только тогда, когда $g_j \in G_q$, то есть объект и груз принадлежат одному и тому же классу. При этом гарантируется, что для каждого груза найдется мобильный объект, который может его перемещать.

Согласно заданию, требуется переместить $|G_m| = P \le L$ грузов $g_1^m, g_2^m, \ldots, g_P^m$ из начальных позиций

$\left(x_{g_1^m}^{(0)}, y_{g_1^m}^{(0)}\right), \left(x_{g_2^m}^{(0)}, y_{g_2^m}^{(0)}\right), \ldots, \left(x_{g_K^m}^{(0)}, y_{g_K^m}^{(0)}\right)$ в позиции

$\left(x_{g_1^m}^{(e)}, y_{g_1^m}^{(e)}\right), \left(x_{g_2^m}^{(e)}, y_{g_2^m}^{(e)}\right), \ldots, \left(x_{g_K^m}^{(e)}, y_{g_K^m}^{(e)}\right)$ соответственно,

так чтобы общее время выполнения задания было минимально. При этом гарантируется, что все $\left(x_{g_1^m}^{(e)}, y_{g_1^m}^{(e)}\right)$ различны и не заняты «неподвижными» грузами (то есть теми, которые перемещать не требуется).

Чтобы формализовать эту задачу, сначала введем несколько определений.

Определение 1. *Заданием* мобильного объекта r_i назовем упорядоченное множество грузов $G_{r_i} = \{g_i^1, g_i^2, \ldots, g_i^k\} \subseteq G_m$, для которого выполняются следующие условия:

1) Грузы перечислены в том порядке, в котором их нужно переместить на конечные позиции;

2) Классы всех грузов совпадают с классом мобильного объекта:

$$r_i \in R_q \Rightarrow \left(\forall g_j \in G_{r_i} : g_j \in G_q\right). \tag{7}$$

3) Каждый подвижный груз принадлежит ровно одному заданию:

$$\forall i, j = \overline{1,K} : \left(i \neq j \Rightarrow G_{r_i} \cap G_{r_j} = \varnothing\right); \tag{8}$$

$$\bigcup_{i=1}^{K} G_{r_i} = G_m. \tag{9}$$

Заметим, что для некоторых мобильных объектов задание может быть пустым.

Определение 2. *Распределением заданий между мобильными объектами* $z(G_m)$ назовем такое разбиение множества G_m на непересекающиеся упорядоченные множества $G_{r_1}, G_{r_2}, \ldots, G_{r_K}$, что G_{r_i} является заданием объекта r_i для всех $i = \overline{1,K}$.

Множество различных распределений заданий между мобильными объектами обозначим через $Z(G_m)$.

Определение 3. *Траекторией* мобильного объекта r_i *согласно заданию* G_{r_i} назовем вектор-функцию $tr\left(G_{r_i}\right) = \left\{\left(x_i^{(0)}, y_i^{(0)}\right), \left(x_i^{(1)}, y_i^{(1)}\right), \ldots, \left(x_i^{(e)}, y_i^{(e)}\right)\right\}$, удовлетворяющую следующим условиям:

1) Если $G_{r_i} = \varnothing$, то $tr\left(G_{r_i}\right) = \left\{\left(x_{r_i}^{(0)}, y_{r_i}^{(0)}\right)\right\}$.

2) Если $G_{r_i} \neq \varnothing$, то траектория объекта содержит последовательность начальных и конечных координат грузов из его задания в порядке их перечисления:

$$\left\{\left(x_{g_i^1}^{(0)}, y_{g_i^1}^{(0)}\right), \left(x_{g_i^1}^{(e)}, y_{g_i^1}^{(e)}\right), \left(x_{g_i^2}^{(0)}, y_{g_i^2}^{(0)}\right), \left(x_{g_i^2}^{(e)}, y_{g_i^2}^{(e)}\right), \ldots, \left(x_{g_i^k}^{(0)}, y_{g_i^k}^{(0)}\right), \left(x_{g_i^k}^{(e)}, y_{g_i^k}^{(e)}\right)\right\} \subseteq tr\left(G_{r_i}\right), \tag{10}$$

причем $\left(x_i^{(0)}, y_i^{(0)}\right) = \left(x_{r_i}^{(0)}, y_{r_i}^{(0)}\right)$ и $\left(x_i^{(e)}, y_i^{(e)}\right) = \left(x_{g_i^k}^{(e)}, y_{g_i^k}^{(e)}\right)$.

3) Расстояние между любыми двумя соседними координатами из $tr\left(G_{r_i}\right)$ равно единице:

$$\left|x_i^{(j)} - x_i^{(j-1)}\right| + \left|y_i^{(j)} - y_i^{(j-1)}\right| = 1\ \forall j = \overline{1,e}. \tag{11}$$

4) Все координаты из траектории, за исключением начальных и конечных координат грузов, принадлежат множеству дорог D.

5) При движении по траектории мобильный объект никогда не движется в запрещенном направлении.

Множество всех возможных траекторий $tr\left(G_{r_i}\right)$ мобильного объекта r_i согласно заданию G_{r_i} обозначим через $TR\left(G_{r_i}\right)$.

Определение 4. *Вектором поломок*

$$b\left(tr\left(G_{r_i}\right)\right) = \left\{b_i^{(0)}, b_i^{(1)}, \ldots, b_i^{(e)}\right\}$$

мобильного объекта r_i, который движется по траектории $tr\left(G_{r_i}\right)$ согласно заданию G_{r_i}, назовем вектор, обладающий следующими свойствами:

1) Размерность вектора $b\left(tr\left(G_{r_i}\right)\right)$ совпадает с длиной траектории $tr\left(G_{r_i}\right)$ мобильного объекта r_i:

$$\left|b\left(tr\left(G_{r_i}\right)\right)\right| = \left|tr\left(G_{r_i}\right)\right|. \tag{12}$$

2) $b_i^{(j)} \in \{0,1\}$, причем $b_i^{(j)} = 1$ в том и только том случае, когда объект r_i сломался в окрестности клетки $\left(x_i^{(j)}, y_i^{(j)}\right)$.

Множество возможных векторов поломок объекта r_i, движущегося по траектории $tr\left(G_{r_i}\right)$ согласно заданию G_{r_i}, обозначим через $B\left(tr\left(G_{r_i}\right)\right)$.

Определение 5. *Путем без ожиданий* мобильного объекта r_i с вектором поломок $b\left(tr\left(G_{r_i}\right)\right)$ назовем вектор $w_{n_i} = w_n\left(b\left(tr\left(G_{r_i}\right)\right)\right)$, который можно получить по следующему алгоритму:

1) Изначально вектор w_{n_i} пуст, индекс $j = 1$.

2) Добавляем в вектор w_{n_i} координату $\left(x_i^{(j)}, y_i^{(j)}\right)$.

3) Если j-й элемент вектора поломок $b\left(tr\left(G_{r_i}\right)\right)$ равен 1, добавляем в вектор w_{n_i} координату $\left(x_i^{(j)}, y_i^{(j)}\right)$ еще t_r раз.

4) Если $\left(x_i^{(j)}, y_i^{(j)}\right) \in H$, добавляем в вектор w_{n_i} координату $\left(x_i^{(j)}, y_i^{(j)}\right)$ еще t_l раз.

5) Если $j = e$, прекращаем выполнение алгоритма, иначе увеличиваем j на единицу и переходим к пункту 2.

Определение 6. Пусть для всех мобильных объектов $r_1, r_2, ..., r_K$ известны их пути без ожиданий $w_{n_1}, w_{n_2}, ..., w_{n_K}$. *Путями с ожиданиями* $w_{s_1}, w_{s_2}, ..., w_{s_K}$ объектов $r_1, r_2, ..., r_K$ соответственно назовем совокупность векторов, которую можно получить согласно следующему алгоритму:

1) Изначально векторы $w_{s_1}, w_{s_2}, ..., w_{s_K}$ пусты, индексы $j_1 = j_2 = ... = j_K = 1$.

2) Полагаем индекс $i = 1$, вводим переменные $f_1 = f_2 = ... = f_K = 0$ и выполняем следующие шаги:

2.1. Если $i > K$, переходим к шагу 3.

2.2. Сравниваем элемент $w_{n_i}\left[j_i\right]$ со всеми элементами $w_{n_p}\left[j_p\right]$, где $p = \overline{1,K}$, $p \neq i$:

– если он совпал с каким-либо их этих элементов и x-координаты (абсциссы) элементов $w_{n_i}\left[j_i\right]$ и $w_{n_p}\left[j_i - 1\right]$ совпадают, добавляем в вектор w_{s_i} элемент $w_{n_i}\left[j_i - 1\right]$;

– иначе добавляем в вектор w_{s_i} элемент $w_{n_i}\left[j_i\right]$ и присваиваем $f_i = 1$.

2.3. Увеличиваем значение i на единицу и переходим к подпункту 2.1.

3) Если $j_i = \left|w_{n_i}\right|$ для всех $i = \overline{1,K}$, то мы уже сформировали все пути с ожиданиями и алгоритм завершается. В противном случае изменяем значения индексов $j_1, j_2, ..., j_K$: если $j_i \neq \left|w_{n_i}\right|$, присваиваем $j_i = j_i + f_i$, иначе j_i остается прежним. Затем переходим к шагу 2.

Заметим, что поскольку векторы поломок случайны, мы не можем заранее посчитать время выполнения задания. В связи с этим предлагается ввести *пороговую вероятность выполнения задания* γ. При этом задание будет считаться выполнимым тогда и только тогда, когда существуют такие траектории мобильных объектов, что объекты, двигаясь по этим траекториям, справятся с заданием за время, не превышающее τ, с вероятностью не менее γ. Тогда формальная модель будет иметь вид:

$$P\left(\max_{i=\overline{1,K}}\left|w_{s_i}\left(w_{n_1}, w_{n_2}, ..., w_{n_k}\right)\right| \leq \tau\right) \to \max_{(tr_1, tr_2, ..., tr_K) \in \Omega}, \quad (13)$$

где $\Omega = \left\{(tr_1, ..., tr_K) : P\left(\max_{i=\overline{1,K}}\left|w_{s_i}\left(w_{n_1}, ..., w_{n_K}\right)\right| \leq \tau\right) > \gamma\right\}$; $w_{n_i} = w_n\left(b\left(tr\left(G_{r_i}\right)\right)\right)$ – путь без ожиданий i-го мобильного объекта; $b\left(tr\left(G_{r_i}\right)\right) \in B\left(tr\left(G_{r_i}\right)\right)$ – случайный вектор поломок i-го объекта; $tr\left(G_{r_i}\right) \in TR\left(G_{r_i}\right)$ – траектория i-го объекта; G_{r_i} – задание i-го объекта, полученное при разбиении на задания $z : G_m \to \left\{G_{r_1}, G_{r_2}, ..., G_{r_K}\right\}$, где $z \in Z$.

4. Разработка методов решения задачи

Решение задачи (13) можно разбить на 3 этапа:

1 этап – нахождение субоптимального разбиения на задания $z^* : G_m \to \left\{G_{r_1}^*, G_{r_2}^*, ..., G_{r_K}^*\right\}$;

2 этап – нахождение субоптимальных траекторий $\left(tr_1^*\left(G_{r_1}^*\right), tr_2^*\left(G_{r_2}^*\right), ..., tr_K^*\left(G_{r_K}^*\right)\right)$ согласно заданиям $G_{r_1}^*, G_{r_2}^*, ..., G_{r_K}^*$ соответственно;

3 этап – проверка того, что мобильные объекты, двигаясь по субоптимальным траекториям, выполнят задание на время τ с вероятностью не менее γ.

Рассмотрим каждый из этапов подробнее.

4.1. Нахождение субоптимального разбиения на задания

Утверждение 1. Мобильные объекты и грузы каждого из T классов можно рассматривать независимо друг от друга.

Действительно, поскольку мобильные объекты могут перемещать только грузы того же класса, данную задачу можно рассматривать как совокупность независимых задач нахождения субоптимальных разбиений на задания $z_q^* : G_m^q \to \left\{G_{r_1^q}^*, ..., G_{r_u^q}^*\right\}$, где G_m^q – множество подвижных грузов класса q, а $r_1^q, ..., r_u^q$ – мобильные объекты класса q.

Таким образом, субоптимальное разбиение на задания $z^* = \bigcup_{q=1}^{T} z_q^*$, и для его нахождения необходимо распределить множества $G_m^1, G_m^2, ..., G_m^T$ подвижных грузов между мобильными объектами из множеств $R_1, R_2, ..., R_T$ соответственно, так чтобы задание каждого объекта соответствовало его субоптимальной траектории.

Для нахождения *оптимального* разбиения на задания необходимо:

1) Перебрать все возможные разбиения на задания между мобильными объектами одного класса;

2) Для каждого из разбиений найти оптимальную траекторию и затем выбрать из них лучшую.

Утверждение 2. Перебор всех разбиений на задания имеет экспоненциальную сложность.

Действительно, пусть в некотором разбиении на задания мобильному объекту r_i^q класса q

соответствуют грузы $g_1^q,...,g_p^q$. Тогда перебор всех разбиений на задания должен включать перебор всех перестановок грузов $g_1^q,...,g_p^q$ (поскольку под заданием понимается *упорядоченное* множество грузов), то есть $p!$ вариантов. Сложность этого перебора уже экспоненциальна [4].

Таким образом, уже при небольших значениях количества мобильных объектов K, количества грузов L и количества классов T *оптимальное* разбиение на задания не может быть найдено за разумное время.

Метод, предлагаемый авторами статьи, позволяет находить *субоптимальное* разбиение на задания за полиномиальное время. Прежде чем описать этот метод, следует ввести несколько утверждений.

Утверждение 3. С увеличением длин траекторий $tr\left(G_{r_i}\right)$ мобильных объектов $r_i\ \left(i=\overline{1,K}\right)$ ожидаемое время выполнения задания увеличивается.

Действительно, чем длиннее траектория мобильного объекта, тем большее расстояние он проходит, и, следовательно, тем больше ожидаемое количество его поломок в пути. Кроме того, с увеличением длины траектории увеличивается также время мобильного объекта в пути.

Таким образом, разбиение на задания $z: G_m \to \left\{G_{r_1},...,G_{r_K}\right\}$ будет субоптимальным, если существуют такие траектории $\left(tr_1\left(G_{r_1}\right),...,tr_K\left(G_{r_K}\right)\right)$, что $W=\max\limits_{i=\overline{1,K}}\left|tr_i\left(G_{r_i}\right)\right|$ минимально среди всех $z\in Z$.

Утверждение 4. Длина траектории $tr\left(G_{r_i}\right)$ мобильного объекта r_i согласно заданию $G_{r_i}=\left\{g_i^1,\ldots,g_i^k\right\}$ равна сумме длин траекторий $tr\left(g_i^j\right)$, $j=\overline{1,K}$:

$$\left|tr\left(G_{r_i}\right)\right|=\sum_{j=1}^{K}\left|tr\left(g_i^j\right)\right|, \qquad (14)$$

где $\left|tr\left(g_i^j\right)\right|$ – длина траектории из клетки с координатами $\left(x_{r_i}^{(j)},y_{r_i}^{(j)}\right)$ до клетки $\left(x_{g_i^j}^{(e)},y_{g_i^j}^{(e)}\right)$ через клетку $\left(x_{g_i^j}^{(0)},y_{g_i^j}^{(0)}\right)$; $\left(x_{r_i}^{(1)},y_{r_i}^{(1)}\right)=\left(x_{r_i}^{(0)},y_{r_i}^{(0)}\right)$ и $\left(x_{r_i}^{(j)},y_{r_i}^{(j)}\right)=$ $=\left(x_{g_i^{j-1}}^{(e)},y_{g_i^{j-1}}^{(e)}\right)$ при $j>1$.

Доказательство утверждения следует из непрерывности траектории мобильного объекта.

Утверждение 5. Длина кратчайшей траектории между любыми двумя клетками не меняется со временем и может быть вычислена за константное время.

Действительно, поскольку территория производственного помещения и правила движения по этой территории не изменяются со временем, длина кратчайшей траектории также остается неизменной. Заметим, что территория состоит из повторяющихся фрагментов размером 6×8 клеток, содержащих 2 вертикальных и 3 горизонтальных участка дороги (рис. 2). Клетки, в которых пересекаются вертикальная и горизонтальная дорога, назовем *перекрестками* (на рис. 2 перекрестки заштрихованы).

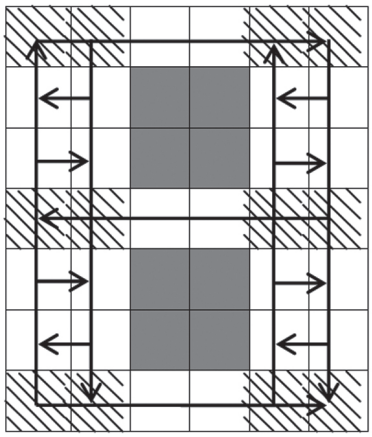

Рис. 2. Повторяющийся фрагмент территории производственного помещения

Расстояние между всеми парами клеток, принадлежащих одному фрагменту, можно найти заранее с помощью алгоритма Флойда-Уоршелла [5] за асимптотику $O\left(V^3\right)$, где V – количество вершин в графе [5]. В данном случае $V=48$ – количество клеток во фрагменте. Если же клетки принадлежат разным фрагментам, путь между ними можно разбить на 3 этапа: путь от 1-й клетки до 1-го перекрестка, путь между перекрестками и путь от последнего перекрестка до 2-й клетки. На 1-м и на 3-м этапах пути клетки, между которыми ищется расстояние, находятся в одном фрагменте. Длину же кратчайшей траектории между перекрестками можно найти за константное время, заметив, что расстояние от любой пары соседних перекрестков до любой другой пары равно манхэттенскому расстоянию между перекрестками из этих пар. Таким образом, заранее просчитав расстояния между всеми парами точек одного фрагмента, кратчайшее расстояние от любой клетки помещения до любой другой можно найти за константное время.

Теперь опишем этапы метода нахождения субоптимального разбиения на задания.

1 этап. Инициализируем $q=1$ – текущий класс мобильных объектов, для которых будет находиться субоптимальное разбиение на задания z_q. Вводим в рассмотрение массив векторов mas_z, где $mas_z[i]$ – текущее задание мобильного объекта r_i (то есть последовательность индексов грузов в том порядке, в котором их нужно забрать этому объекту), и массив целых чисел $state$, где $state[i]=0$, если мобильный объект r_i в данный момент не выполняет задание, и $state[i]=1$, если выполняет. Вводим также массив M_r, такой что $M_r[i]$ – минимальное время, за которое мобильный объект r_i может завершить задание $mas_z[i]$, и список S_{g_m},

содержащий индексы всех подвижных грузов, которые еще не были перемещены. Изначально массив mas_z пуст, массивы *state* и M_r заполнены нулями, а список S_{g_m} содержит индексы всех грузов из G_m. Пусть *curtime* – время, прошедшее с начала выполнения заданий мобильными объектами. Оно также изначально равно 0.

2 этап. Если $g_{m_i} \notin G_q \ \forall j \in S_{g_m}$, увеличиваем q на единицу. Далее для всех мобильных объектов $r_i \in R_q$, выполняющих задание, находим длину кратчайшей траектории l_i, по которой должен двигаться этот мобильный объект для завершения текущего задания (то есть для перемещения текущего объекта в конечную точку). Для всех «свободных» мобильных объектов $r_j \in R_q$ находим длину кратчайшей траектории p_j, по которой должен двигаться этот мобильный объект для выполнения какого-либо задания (то есть наименьшее расстояние, которое нужно пройти объекту, чтобы переместить в конечную точку один из незадействованных до сих пор грузов). Пусть $\bar{p} = \min\limits_j p_j$.

3 этап. Строим двудольный граф, вершинами левой доли которого являются мобильные объекты $r_i \in R_q$, для которых выполняется либо $state[i] = 0$, либо $state[i] = 1$ и $l_i < \bar{p}$, а вершинами правой доли – все грузы класса q из списка S_{g_m}. Длина ребра между i-й вершиной левой доли и j-й вершиной правой доли вычисляется как длина кратчайшей траектории, по которой должен двигаться мобильный объект r_i для того чтобы переместить груз g_{m_j} в его конечную точку. При этом если $state[i] = 1$, в эту траекторию также включается траектория l_i (то есть мобильный объект сначала должен завершить текущее задание, а потом начинать следующее).

4 этап. С помощью венгерского алгоритма решения задачи о назначениях [6] находим такое распределение грузов между мобильными объектами $\left(r_{a_1} \to g_{b_1}, r_{a_2} \to g_{b_2}, ..., r_{a_t} \to g_{b_t}\right)$, что сумма кратчайших траекторий, необходимых этим объектам для перемещения соответствующих объектов в необходимые клетки, минимальна по всем возможным распределениям. Добавляем в конец каждого из векторов $mas_z[a_i]$ соответствующий индекс b_i, присваиваем $state[a_i] = 1$, удаляем индексы $b_1, b_2, ..., b_t$ из списка S_{g_m} и прибавляем к каждому из элементов $M_r[a_i]$ значения $t\left(r_{a_i}, g_{b_j}\right)$, где $t\left(r_i, g_j\right)$ – минимальное время, необходимое мобильному объекту r_i для перемещения груза g_j после окончания перемещения предыдущего груза из его задания.

5 этап. Если список S_{g_m} пуст, завершаем работу метода. Иначе присваиваем $curtime = \min\limits_{i: r_i \in R_q} M_r[i]$, а затем для тех мобильных объектов r_i, у которых $M_r[i] = curtime$, присваиваем $state[i] = 0$ и переходим ко 2-му этапу.

В результате работы данного метода массив векторов mas_z будет содержать субоптимальное разбиение на задания. Оценим сложность работы данного метода. Легко показать, что наибольшую асимптотику работы имеет венгерский алгоритм, который выполняется на 4-м этапе, так что выполнение остальных этапов метода не влияет на общую сложность. Рассмотрим j-ю итерацию метода для класса q. Пусть в двудольном графе, построенном на 3-м этапе, количество вершин в левой доле равно K_{qj}, а в правой – L_{qj}. Случай $K_{qj} \ge L_{qj}$ возможен только на последней итерации для данного класса, так как в этом случае каждый из оставшихся грузов будет перемещен каким-то мобильным объектом; при этом сложность работы j-й итерации 2-го этапа метода, согласно асимптотике венгерского алгоритма, будет равна $O\left(K_{qi} \cdot L_{qi}^2\right)$. Поскольку $\sum\limits_{q=1}^{T} K_{qi} \cdot L_{qi}^2 \le \sum\limits_{q=1}^{T} K_q \cdot L_q^2 \le \ \le \sum\limits_{q=1}^{T} K_q \cdot \left(\sum\limits_{q=1}^{T} L_q\right)^2 = KL^2$, где K_q и L_q – соответственно количество мобильных объектов и грузов класса q, то сложность работы этой части метода равна $O\left(KL^2\right)$.

Во всех остальных случаях $K_{qj} < L_{qj}$, поэтому сложность j-й итерации будет равна $O\left(K_{qj}^2 \cdot L_{qj}\right)$. Заметим, что поскольку на каждой итерации перемещенные грузы удаляются из списка, то $\sum\limits_{j=1}^{d_q} L_{qj} \le L_q$, где d_q – количество итераций для класса q, для которых $K_{qj} < L_{qj}$. Так как

$$\sum_{j=1}^{d_q} K_{qj}^2 \cdot L_{qj} < \sum_{j=1}^{d_q} L_{qj}^3 \le \left(\sum_{j=1}^{d_q} L_{qj}\right)^3 \le L_q^3 \qquad (15)$$

и

$$\sum_{q=1}^{T} L_q^3 \le \left(\sum_{q=1}^{T} L_q\right)^3 = L^3, \qquad (16)$$

то сложность выполнения всех итераций метода, для которых $K_{qj} < L_{qj}$, равна $O\left(L^3\right)$, и общая сложность метода равна $O\left(KL^2 + L^3\right) = O\left(\max\left(K, L\right)^3\right)$.

4.2. Нахождение субоптимальных траекторий

Итак, в результате 1-го этапа работы метода для каждого мобильного объекта r_i был получен список $S = \left\{(x_{1i}, y_{1i}), (x_{2i}, y_{2i}), ..., (x_{qi}, y_{qi})\right\}$ клеток, которые он должен посетить. Согласно утверждениям 3 и 4, чтобы траектория была субоптимальной, объект r_i между любыми двумя соседними клетками из списка S должен идти по кратчайшему пути.

Таким образом, нахождение *субоптимальных* траекторий сводится к нахождению одного из кратчайших путей между двумя произвольными клетками (x_1, y_1) и (x_2, y_2).

Построим ориентированный граф, в котором вершинами будут клетки, а ребрами – разрешенные переходы между соседними клетками. Тогда кратчайший путь между любыми двумя клетками

можно найти с помощью алгоритма поиска в ширину с запоминанием пути. Применяя этот алгоритм для каждого мобильного объекта и для всех участков пути, получим траектории, которые будут кратчайшими и, следовательно, субоптимальными.

Оценим сложность алгоритма. Очевидно, количество вершин в графе будет равно $V = NM$. Поскольку из каждой вершины выходит не более 4 ребер, то количество ребер $E \le 4V = 4NM$. Сложность алгоритма поиска в ширину – $O(E)$ [5], в данном случае – $O(NM)$. Заметим, что каждые два вызова алгоритма соответствуют перемещению одного груза в нужную клетку; соответственно, общее количество вызовов алгоритма – $O(L)$. Итого общая сложность нахождения траекторий для всех мобильных объектов – $O(NML)$.

4.3. Проверка выполнимости задания.

После того, как были получены траектории, по которым должны двигаться мобильные объекты, следует проверить, удовлетворяют ли они ограничениям задачи. Поскольку перебрать все возможные варианты движения и поломок мобильных объектов за разумное время не представляется возможным, вероятность того, что задание будет выполнено, можно оценить с помощью метода Монте-Карло [7].

Сначала выберем число испытаний Q. С одной стороны, оно должно быть весьма велико, чтобы оценка вероятности была достаточно точной, а с другой – слишком большое значение Q приведет к тому, что проверка будет выполняться очень долго. Рекомендуется выбрать Q в пределах от 1000 до 10000.

Теперь необходимо Q раз промоделировать процесс движения мобильных объектов по траекториям согласно правилам, рассмотренным в предыдущем разделе. При этом вместо случайной величины ω интервала между поломками для каждого мобильного объекта генерируются ее реализации, а роль перемещения в ближайшую клетку при поломке играет округление текущих координат мобильного объекта до ближайшего целого.

Когда движение мобильных объектов по траекториям окончено, необходимо сравнить время T, которое прошло с момента начала выполнения задания, с требуемым временем τ. Если $T \le \tau$, задание успешно выполнено и количество успешных испытаний необходимо увеличить на единицу; иначе задание считается невыполненным.

Пусть в результате моделирования процесса из Q испытаний успехом окончилось U. Тогда если $\frac{U}{Q} \ge \gamma$, то задача решена – мобильные объекты, двигаясь по траекториям, найденным на предыдущем этапе, с вероятностью γ выполнят задание вовремя. Если же $\frac{U}{Q} < \gamma$, считаем, что задача не может быть решена при заданных ограничениях, и сообщаем, что задачу следует модифицировать.

Оценим сложность выполнения данного этапа. Заметим, что длина траектории мобильного объекта между двумя клетками, которые ему необходимо посетить, не превышает $N + M$. Тогда длина пути, который необходимо пройти, чтобы переместить груз в нужную клетку, не превышает $2(N + M)$, и суммарная длина всех траекторий не больше $2L(N + M)$. Таким образом, при достаточно высокой надежности мобильных объектов (т.е. количество их поломок в пути невелико) и относительно небольших значениях отношений t_l / t_0 и t_r / t_0, моделирование одного испытания имеет сложность $O\big(L(N+M)\big) = O\big(L \cdot \max(N, M)\big)$, а моделирование всех испытаний – соответственно $O\big(QL \cdot \max(N, M)\big)$.

Таким образом, общая сложность метода субоптимальной маршрутизации мобильных объектов равна $O\Big(\max(K, L)^3 + NML + QL \cdot \max(M, N)\Big)$.

5. Программная реализация метода

Описанный метод программно реализован на языке Java; в среде IntelliJ Idea разработано графическое приложение, иллюстрирующее движение мобильных объектов по найденным траекториям.

UML-диаграмма классов приложения приведена на рис. 3. Опишем подробнее каждый из приведенных классов.

InputParametersFrame – класс диалогового окна, в которое вводятся входные параметры модели: размеры поля N и M, количество мобильных объектов K, их координаты и классы; количество грузов, их координаты и классы; новые координаты грузов, которые необходимо переместить; максимальное время выполнения задания T, пороговая вероятность γ и параметр показательного распределения λ. Метод CheckParameters проверяет введенные параметры на корректность.

ParametersContainer – класс, в котором хранятся входные параметры, введенные пользователем. К этим данным предоставляется доступ другим классам.

FieldFrame – диалоговое окно, содержащее кнопки меню «Выход», «Изменить параметры» и «Пауза/продолжить», позволяющие управлять соответственно: завершить моделирование перемещения мобильных объектов и выйти из программы; завершить моделирование и изменить входные параметры; прервать или возобновить моделирование.

FieldPanel – поле, расположенное внутри диалогового окна *FieldFrame*, в котором моделируется перемещение мобильных объектов по территории производственного помещения.

GeneralObject – обобщенный класс, содержащий поля и методы, общие для мобильного объекта и

груза – текущие координаты, цвет и размер на экране, номер класса, а также метод draw(), рисующий данный объект на экране, и метод move(), передвигающий его.

MobileObject – класс мобильного объекта, который наследует класс *GeneralObject*. В нем перегружаются методы draw() и move(), а также вводятся такие поля, как траектория мобильного объекта и его состояние (движется, загружается, разгружается, ожидает или сломался).

Load – класс груза, который также наследует класс *GeneralObject*. В нем также перегружены методы draw() и move(), и, кроме того, введены поля, показывающие, является ли данный груз подвижным и перемещается ли он в данный момент.

TrajectoryGenerator – класс, в котором реализован вышеописанный метод нахождения субоптимальных траекторий мобильных объектов. В методе findDistance определяется кратчайший путь между двумя клетками, как описано в утверждении 5; методы getSchedule, getTrajectories и checkTrajectories реализуются соответственно: 1-й, 2-й и 3-й этапы метода – получение разбиения на задания, получение траекторий мобильных объектов и проверка полученных траекторий. В результате, если решение существует, инициализируются поля trajectory всех мобильных объектов и внутри поля FieldPanel моделируется их перемещение согласно полученным траекториям. Если же решения не существует, выводится сообщение об этом и пользователю предлагается модифицировать входные параметры.

Поскольку предложенный метод имеет кубическую сложность, на входные параметры были введены следующие ограничения: длина и ширина поля $N, M \le 100$; количество мобильных объектов и грузов $K, L \le 200$.

Программная реализация метода подтвердила его эффективность.

Выводы

Усовершенствована модель перемещения группы сложных объектов по территории производственного помещения, которая, в отличие от существующих, минимизирует взаимодействие этих объектов, что позволяет повысить качество управления.

Получил дальнейшее развитие метод маршрутизации группы сложных объектов, который, в отличие от существующих, находит их субоптимальные траектории за время, полиномиальное относительно входных параметров, что позволяет минимизировать вычислительную сложность системы.

Усовершенствован венгерский алгоритм решения задачи о назначениях для случая, когда каждый процесс может выполнять несколько видов работ. При этом полученное решение может оказаться неоптимальным, однако асимптотика метода остается кубической.

Программно реализован метод субоптимальной маршрутизации сложных объектов, который работает за кубическую асимптотику.

Перспективным направлением дальнейших исследований является использование нечеткой логики и знание-ориентированных технологий.

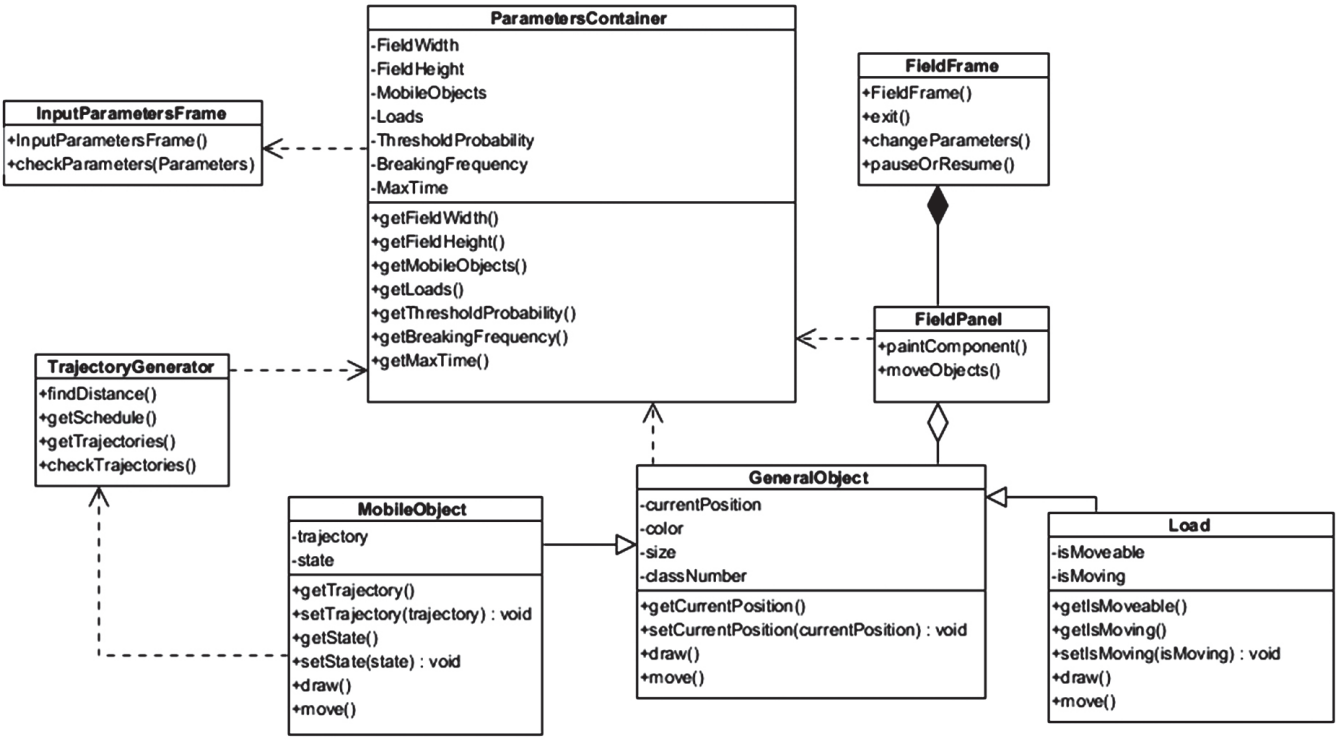

Рис. 3. UML-диаграмма классов приложения

Список литературы: 1. Управление ГПС: Модели и алгоритмы [Текст] / Под общ. ред. акад. С. В. Емельянова. – М.: Машиностроение. 1987. – 368 с., ил. **2.** *Даринцев, О. В.* Нейросетевой алгоритм планирования траектории для группы мобильных роботов [Текст] / О. В. Даринцев, А. Б. Мигранов, Б. С. Юдинцев // Искусственный интеллект. – 2011. – №1.– С. 154–160. **3.** Транспортные роботы для гибких производственных систем [Текст] / Н. М. Довбня, А. А. Халфен, И. В. Яковлев. Л.: ЛДНТП, 1988 г. – 233 с.: ил. **4.** *Кузюрин, Н. Н.* Эффективные алгоритмы и сложность вычислений [Текст]: учеб. пособие для вузов / Н. Н. Кузюрин, С. А. Фомин. – М., 2008. – 317 с. **5.** Алгоритмы: построение и анализ [Текст]: пер. с англ. / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – М.: Вильямс, 2006. – 1296 с. **6.** *Ahuja, R. K.* Network flows: theory, algorithms, and applications. / R. K. Ahuja, T. L. Magnanti, J. B. Orlin. – New Jersey, Prentice-Hall, 1993. **7.** *Войтишек А. В.* Основы метода Монте-Карло [Текст]: Учеб. пособие / А. В. Войтишек. – Новосибирск: Новосиб. Гос. ун-т, 2010. – 108 с.

Поступила в редколлегию 26.10.2012

УДК 004.93

Стохастичні моделі і методи субоптимальної маршрутизації складних об'єктів / А.Д. Дрюк, Є.І. Кучеренко // Біоніка інтелекту: наук.-техн. журнал. – 2013. – № 1 (80). – С. 45-53.

У статті запропоновано розроблену авторами модель переміщення групи мобільних об'єктів згідно з заданими правилами. Для даної моделі розглянуто метод субоптімальної маршрутизації мобільних об'єктів. Стверджується, що цей метод працює за поліноміальний час і дозволяє знаходити рішення задачі переміщення грузів по виробничій території за фіксований час.

Іл. 3. Бібліогр.: 7 найм.

UDK 004.93

Stochastic models and methods of suboptimal complex objects routing / O.D. Driuk, Ye.I. Kucherenko // Bionics of Intelligense: Sci. Mag. – 2013. – № 1 (80). – P. 45-53.

In article the model of moving of mobile objects according to definite rules developed by authors of article is proposed. For given model the method of suboptimal mobile objects routing is considered. This method is approved to work for polynomial time and to find the solution of problem of moving loads through production area for fixed period of time.

Fig. 3. Ref.: 7 items.