УДК 004.01

Степанова Т.А.
студент

Кацупеев А.А.
канд. тех. наук, доцент

Южно-Российский государственный политехнический университет (НПИ) имени М.И. Платова
Россия, г. Новочеркасск

ЗАДАЧА ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ПРОЦЕССОВ В СИППР «ЭЛЕКТРОННАЯ ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ»

Системы поддержки принятия решений очень востребованы на крупных предприятиях, где речь идет о работе с большими объемами данных. Разносторонний подход к управлению их функционированием дает толчок к развитию предприятия и отрасли в целом. В данной статье проведено исследование производственной ситуации, возникающей в системе при распределении работ по информационным блокам. При этом были задействованы методы, применяемые обычно при решении задач дискретного программирования. В процессе была сформирована математическая постановка задачи, разработан подробный алгоритм метода решения и структурная схема, а также программа для выполнения автоматических расчетов.

Ключевые слова*: информационная система, система поддержки принятия решений, эксплуатационная документация, нештатные ситуации, информационный блок, распределение процессов, дискретное программирование, алгоритм, метод ветвей и границ, анализ данных, распределение ресурсов, оперативное планирование.*

Оперативное планирование на предприятии в настоящее время не может обойтись без грамотного плана распределения ресурсов. В таких отраслях, как приборостроение, авиационная и космическая промышленность, например, для бортовых механик важна задача разделения определенного числа работ по отслеживающим приборам. Для системы поддержки принятия решений, отвечающей за анализ выполнения тестовых нештатных ситуаций, внедрение решения этой задачи позволяет получить существенный экономический эффект без лишних затрат на дополнительное оборудование, а также более выраженную конкретность при работе с данными [1; 10]. На концептуальном и математическом уровнях ее можно свести к дискретному программиро-

© Степанова Т.А., Кацупеев А.А., 2022

ванию, а именно к обобщенной минимаксной задаче о назначениях. Для решения целесообразно использовать различные точные алгоритмы, которые базируются на методе ветвей и границ, а также методе динамического программирования [2; 3; 5; 6].

Исходную задачу можно сформулировать как: имеется множество процессов (работ) $A = \{ A_1, ..., A_i, ..., A_m\}$, множество блоков информационной системы $B = \{ B_1, ..., B_j, ..., B_n\}$, а также матрица затрат времени $T=||t_{ij}||$, где t_{ij} - время выполнения i-го процесса *j-м* блоком, $i=1,m;\ j= 1,n$. При этом каждый из блоков может выполнять произвольное количество процессов, а каждый процесс может отслеживаться только одним блоком. Тогда необходимо распределить выполнение процессов таким образом, чтобы все они были закончены в наиболее ранние сроки.

Для этого должны быть введены переменные:

S = {1,…,s,…,s0} – множество вариантов распределения процессов по блокам;

$N_j(s)$ – то множество процессов, которые назначены j-му блоку в s-ом варианте;

$T_j\ (s) = \sum_{i \in N\ j\ (s)} t_{ij}$ – время загруженности j-го блока в s-м варианте распределения процессов;

$T(s) = minT_i\ (s)$ – время окончания процессов s-го варианта.

Тогда из множества вариантов S необходимо выбрать такой вариант $l \in S$, для которого $T(l) = \min T(s)$.

Этап 1. Этот этап определяет множество G0 – множество всех вариантов распределения работ по исполнителям (рис. 1), находится нижняя оценка этого множества $\xi(G_0) = \max(\Delta_1, \Delta_2)$, где Δ_1, и Δ_2 – частные оценки.

Вариант s = 1 : все процессы назначены первому блоку,
Вариант s = 2: первый процесс назначен второму блоку,
все остальные процессы - первому блоку,
Вариант s = s0: все процессы назначены n-му блоку.

Рис. 1. Варианты распределения для исходного множества G_0

Первая частная оценка Δ_1 вычисляется следующим образом : для каждого из процессов нужно найти минимальное время его выполнения: $\tau_i = \max t_{ij}, i = \overline{1, m}$. Тогда

$$\Delta_1 = \max \tau_i, i = \overline{1, m}..$$

Вторую частную оценку Δ_2 можно найти из соотношения: $\Delta_2 = \frac{1}{n}\sum_{i=1}^{m} \tau\ i$.

Этап 2. Исходное множество G_0 разбивается на n непересекающихся подмножеств: G_1^1, G_2^1, …, G_y^1, …, G_n^1. Количество подмножеств равно числу блоков n. Для каждого из подмножеств необходимо найти нижнюю оценку $\xi(G_y^1,)$ (рис. 2).

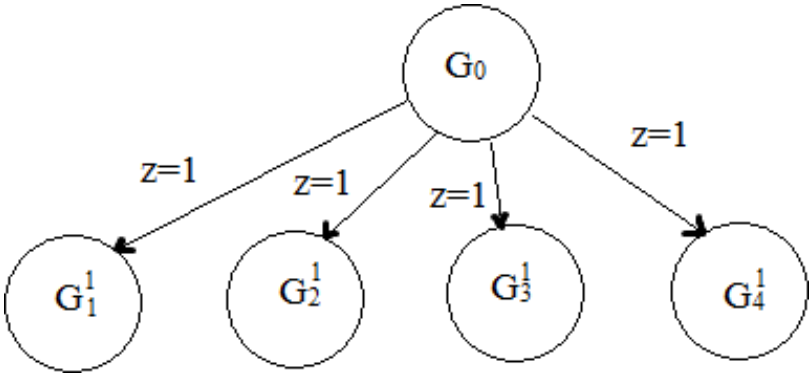

Рис. 2. Граф ветвления исходного множества G_0

Далее выбирается распределяемая работа, то есть та, которая имеет номер z = 1. Множество G_1^1 характеризуется тем, что первый процесс назначен первому информационному блоку; множество G_2^1- тем, что первый процесс назначен второму блоку и т.д. Для каждого из подмножеств далее определяется значение элементов, которые входят в множество $N_j(G_1^1)$, где $N_j(G_1^1)$ – это те процессы, которые назначены первому блоку в подмножестве решений G_1^1.

Очевидно, что

$$N_1(G_1^1) = \{1\}, N_1(G_2^1) = \{0\},\ N_1(G_y^1)=\{0\},\quad N_1(G_n^1) = \{0\},$$

$$N_2(G_1^1) = \{0\},\ N_2(G_2^1) = \{1\}, N_2(G_v^1) = \{0\},\ N_2(G_n^1) = \{0\}, N_j(G_1^1) = \{0\}, N_j(G_2^1) = \{0\},$$

$$N_j(G_v^1) = \{1\}, N_j(G_n^1) = \{0\}, N_n(G_1^1) = \{0\},\ N_n(G_2^1) = \{0\},\ N_n(G_v^1) = \{0\}, N_n(G_n^1) = \{1\},$$

Для каждого из подмножеств G_y^1 находим нижнюю оценку $\xi(G_y^1)$.

В частности, необходимо рассмотреть нахождение нижней оценки для произвольного подмножества G_y^1. Тогда $\xi(G_y^1) = max(\Delta_1,\Delta_2,\Delta_3)$.

Для вычисления нижней оценки Δ1 используются следующие соотношения:

$$\tau_i = \min t_{ij}\ ,\ \Delta_1 = \max\tau_i,\ i = 2, m.$$

Так же определение нижних оценок Δ_2 и Δ_3 составляется как вычисление загрузки *i*-го исполнителя для подмножества G_v^1, т.е. $T_{j1}(G_v^1) = \sum_{i \in N_j(G1)} t_{ij}$, а после этого полученные времена загрузки блоков должны быть упорядочены в порядке возрастания, т.е. мы получаем последовательность $T_{j1}(G_v^1) \leq T_{j2}(G_v^1) \leq \cdots \leq T_{jn}(G_v^1)$, и, наконец, нижняя оценка Δ_3 находится как $\Delta_3 = T_{jn}(G_y^1)$.

Этап 3. Из имеющихся конкурирующих подмножеств необходимо выбрать перспективное.

В качестве него берется то, которое имеет минимальную нижнюю оценку. Таким же образом продолжается выполнение следующей итерации, в которую входят второй и третий этапы.

Создаваемый программный продукт, в свою очередь, должен отвечать определенным условиям, таким, как возможность ввода и редактирования количества процессов, блоков, ввода матрицы времени [9]. Программа должна осуществлять вывод оптимального распределения процессов и времени выполнения каждого из них [7]. В ходе разработки были созданы такие структуры данных, как *Matrix* (должна хранить матрицу состояний), *WorkSet* (хранит информацию обо всех подмножествах), и *Raspr* (хранит информацию о распределенных процессах).

Поля структуры Matrix представлены в таблице 1.

Таблица 1

Структура Matrix

Поле	Тип	Назначение
S	integer	Массив, заключающий в себе расстояние между затратами времени

Поля WorkSet представлены в таблице 2.

Таблица 2

Структура WorkSet

Поле	Тип	Назначение
Ksi	integer	Нижняя оценка множества
N	Unsigned integer	Количество распределенных процессов
SS	Raspr	Матрица распределенных процессов
num	Unsigned integer	Количество перспективных ветвей
first	*WorkSet	Указатель на первый элемент списка
last	*WorkSet	Указатель на последний элемент списка
next	*WorkSet	Указатель на следующий элемент списка

Поля Raspr описаны в таблице 3.

Таблица 3

Структура Raspr

Поле	Тип	Назначение
p	Unsigned integer	Инфоблок, на который распределен процесс
r	Unsigned integer	Уже распределенные процессы

На рисунке 3 представлена блок-схема алгоритма программы, модулирующей задачу и решающей ее методом ветвей и границ.

Рис. 3. Алгоритм реализации автоматизированного решения

Реализация программы была выполнена в интегрированном пакете C++ Builder 6.0. Главное окно программы и результат работы при заданных значениях (на примере 3 инфоблоков, 5 процессов и выбранной матрице времени) показаны на рис. 4–5 [4; 8].

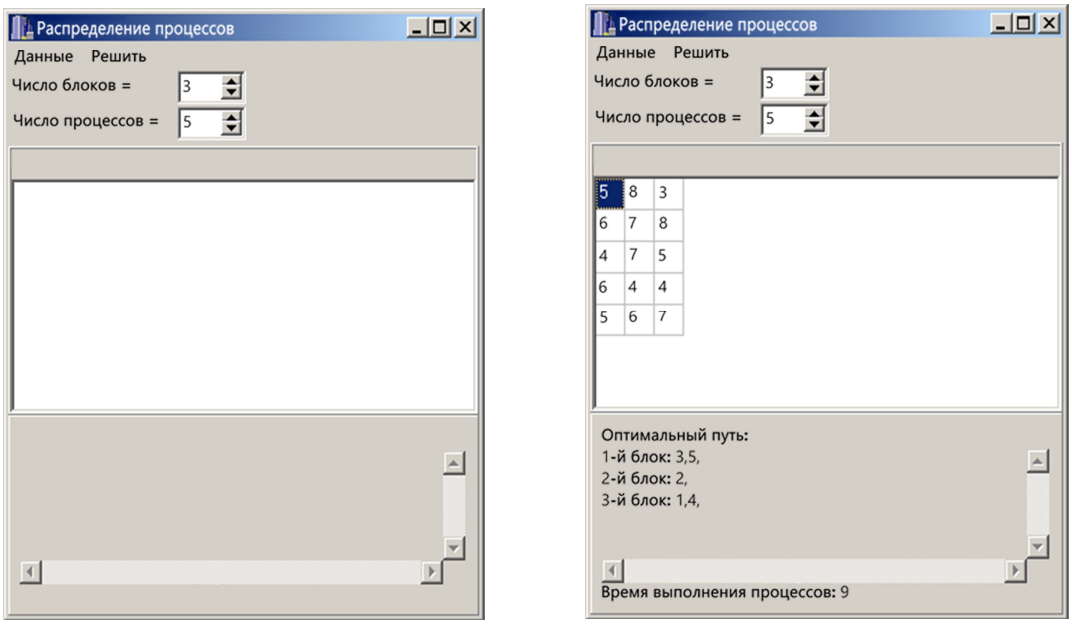

Рис. 4–5. Внешний вид и работа программы

После ввода пользователем всех необходимых параметров задачи выбирается пункт «Решить», и программа выведет все полученные значения внизу главного окна.

Если при вводе параметров будет допущено нарушение формата, программа даст соответствующее сообщение и предложит пользователю заменить неподходящее значение на допустимое.

При ручной проверке соответствия результатов строится вспомогательная таблица, содержащая строки τ_i z и $\sum_{i=p}^{z} \tau i$, находится нижняя оценка для каждого подмножества ($\xi(G_1^1)= \xi(G_2^1)= \xi(G_3^1)= \max(\Delta_1,\Delta_2,\Delta_3)$), выбираются перспективные вершины в процессе нескольких итераций, после чего получится дерево решений следующего вида (рис. 6):

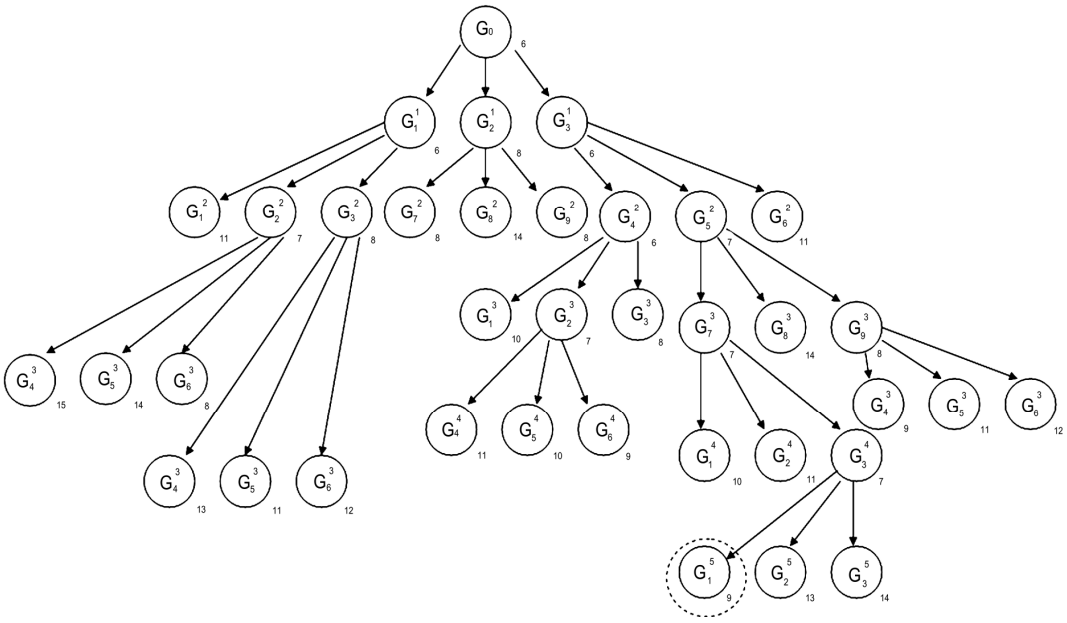

Рис. 6. Дерево решений

Спускаясь по дереву, можно увидеть, что распределение на 1-й блок 3 и 5 процесса, на 2-й блок 2-го процесса, на 3-й блок 1 и 4 процесса является оптимальным. При этом общая продолжительность процессов равна 9, что доказывает корректность работы программы.

С точки зрения ее применения в СИППР «Электронная эксплуатационная документация», стоит рассмотреть функциональную структуру данной системы (рис. 7). Для функции «Анализ данных оператора» выделено два блока: «редактирование статуса» и «вывод журнала действий». Оба информационных блока предоставляют пользователю работу с хранящимися в базе данными оператора. Для функции «Моделирование нештатных ситуаций» выделено три блока: «выбор нештатной ситуации», «переход по пунктам сценария» и «вычисление результатов анализа». Все они задействуют аналитические возможности системы для определения конечного результата. Распределение параллельных процессов нескольких одновременно участвующих в работе операторов, учитывая полученные результаты, может проводиться наиболее эффективно.

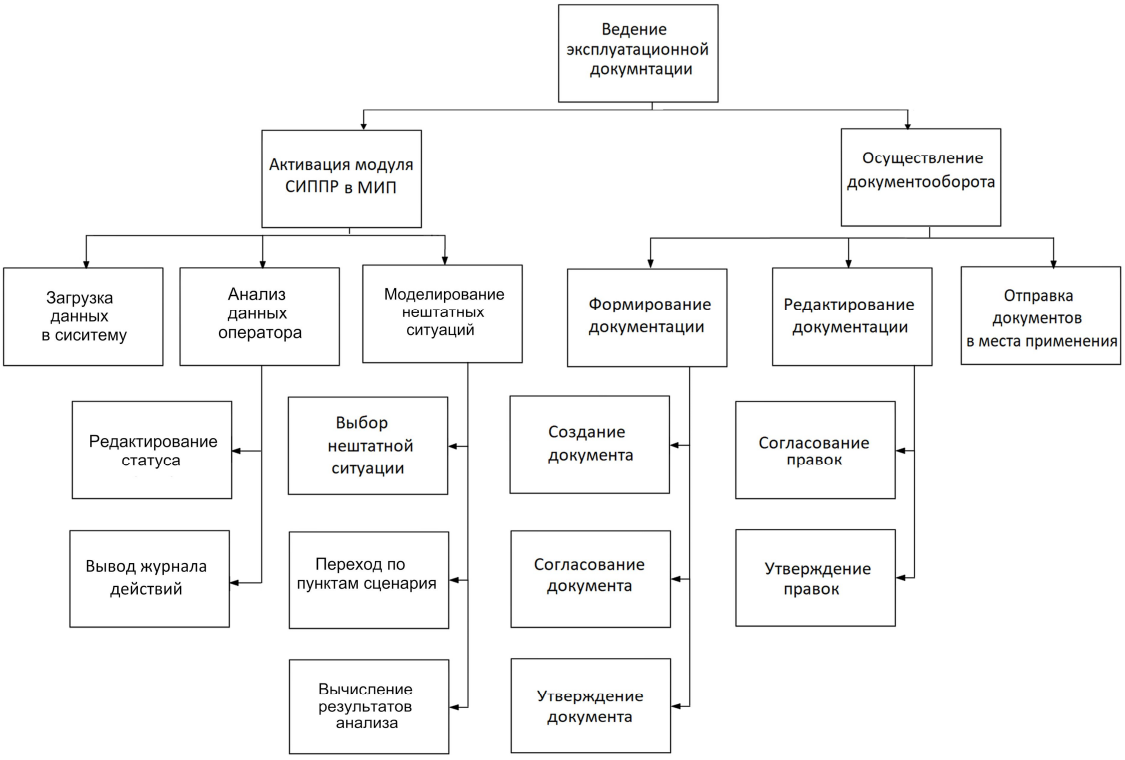

Рис. 7. Функциональная структура СИППР

Таким образом, использование метода дискретного программирования в реализованной программе способствует улучшению качества системы и ускорению ее работы.

СПИСОК ЛИТЕРАТУРЫ

1. Антонов А.В. Системный анализ: учебник для вузов. – 2014. –238 с.

2. Струченков В.И. Дискретная оптимизация. Модели, методы, алгоритмы решения прикладных задач. – М.: СОЛОН-ПРЕСС, 2016. – 192 с.

3. Павловская Т.А. C# программирование на языке высокого уровня: учебник для вузов. – СПб.: Питер, 2014. – 432 с.

4. Албахари Дж., Албахари Б. C# 5.0: справочник. Полное описание языка. – 2014. – 1008 с.

5. Окулов С., Пестов О. Динамическое программирование. – М.: Бином. Лаборатория знаний, 2015. – 296 с.

6. Панфилов А.Н., Зуев В.А., Скоба А.Н. Методы и средства проектирования информационных систем и технологий // Юж.-Рос. гос. политехн. ун-т (НПИ) имени М.И. Платова. – Новочеркасск: Лик, 2016. – 60 с.

7. Электронный учебник «Экономико-математические методы» [электрон. текстовые данные]. – Режим доступа: http://www.math.mrsu.ru/text/courses/method/metod_vetvei_i_granic.htm.

8. Советов Б.Я. Архитектура информационных систем: учебник для студ. учреждений – М.: Изд. центр «Академия», 2012. – 496 с.

9. Радоуцкий В.Ю, Шульженко В.Н., Носатова Е.А. Основы научных исследований: учеб. пос. – Бел-город: Изд-во БГТУ им. В.Г. Шухова, 2008. – 133 с.

10. Сараев А.Д., Щербина О.А. Системный анализ и современные информационные технологии // Труды Крымской Академии наук. – Симферополь: СОНАТ, 2009. – 136 с.

Stepanova T.A.
student

Katsupeev A.A.
Candidate of Engineering Sciences, Associate Professor

South Russian State Polytechnic University (NPI) named after M.I. Platova
Russia, Novocherkassk

THE PROBLEM OF OPTIMAL DISTRIBUTION OF PROCESSES IN THE DISP “ELECTRONIC ACCOMPANYING DOCUMENTATION»

Decision support systems are in great demand in large enterprises, where it comes to working with large amounts of data. A versatile approach to managing their operation gives impetus to the development of the enterprise and the industry as a whole. In this article, a study was made of the production situation that occurs in the system when the work is distributed among information blocks. At the same time, methods were used that are usually used in solving discrete programming problems. In the process, a mathematical formulation of the problem was formed, a detailed algorithm for the solution method and a structural diagram were developed, as well as a program for performing automatic calculations.

Key words: *information system, decision support system, operational documentation, emergency situations, information block, process distribution, discrete programming, algorithm, branch and bound method, data analysis, resource allocation, operational planning.*