

УДК 519.816+338.24

СРЕДСТВА ВЕКТОРНОЙ ОПТИМИЗАЦИИ СТРУКТУРЫ
КЛАСТЕРОВ В СРЕДЕ *MATHCAD* НА ОСНОВЕ МЕТОДА *STEM*

К.А. Булыгин, В.Ю. Пинкевич, В.А. Богатырев

Предложены средства решения задач векторной оптимизации в системе компьютерной математики Mathcad на основе метода STEM. Рассмотрено применение предлагаемых средств для оптимизации структуры кластера.

Ключевые слова: векторная оптимизация, STEM, принятие решений, кластер, компьютерная математика, Mathcad.

Введение

Проектирование компьютерных систем кластерной архитектуры связано с обеспечением их высокой надежности и производительности при низких затратах на построение и эксплуатацию системы, что обуславливает необходимость решения задач векторной оптимизации [1].

Эффективность векторной оптимизации во многом зависит как от выбора метода поиска решений, так и от используемых для этого инструментальных средств. В качестве инструментальных средств оптимизации может использоваться система компьютерной математики Mathcad, включающая встроенные функции оптимизации *Minimize* (f, x_1, x_2, \dots) / *Maximize* (f, x_1, x_2, \dots) [2]. Сведение векторной задачи оптимизации к скалярной на основе метода отклонения от идеала, мультипликативного или аддитивного критерия [1], программная реализация которых в среде Mathcad предложена в [3], сопряжена с субъективностью задания весов (важности) частных критериев. Формально определить веса частных критериев позволяет известный метод STEM, предполагающий выполнение многоэтапной процедуры [1]. В настоящей работе решается задача расширения возможностей системы Mathcad в результате программирования набора средств (функций) поддержки векторной оптимизации на основе метода STEM.

Постановка задачи оптимизации кластера

В качестве объекта оптимизации рассмотрим многоуровневую компьютерную систему кластерной архитектуры с резервированием узлов на каждом уровне (рисунок) [4, 5].

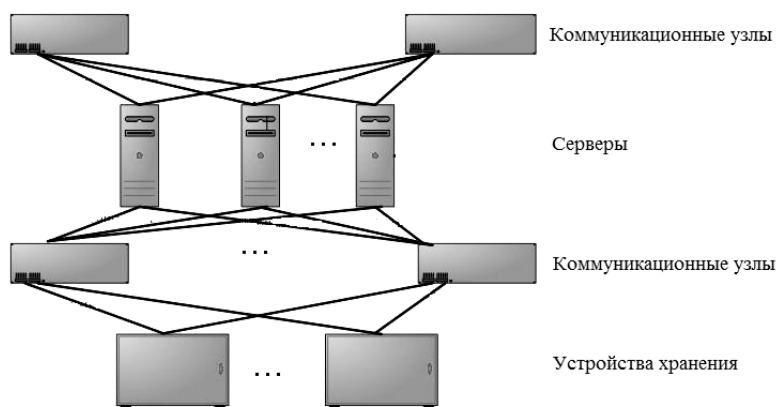


Рисунок. Структура кластерной системы

Требуется найти кратность резервирования (число узлов на каждом уровне) m_1, m_2, \dots, m_M , при которой обеспечивается максимум надежности системы, минимум среднего времени пребывания в ней заявок, максимум интенсивности входного потока заявок, который система выдерживает в стационарном режиме, и минимум стоимости. Кроме того, стоимость реализации системы не должна превышать C_0 . Таким образом, $U(m_1, m_2, \dots, m_M) \rightarrow \min$, $\Lambda(m_1, m_2, \dots, m_M) \rightarrow \max$ и $C(m_1, m_2, \dots, m_M) \rightarrow \min$ при

$C(m_1, m_2, \dots, m_M) = \sum_{i=1}^M m_i c_i \leq C_0$. Возможна постановка задачи минимизации стоимости при ограничении

остальных показателей. Задачи векторной оптимизации кластеров в различных постановках решались в [5–9]. При оптимизации будем считать заданными: интенсивность входного потока заявок λ ; вероятность безотказной работы узлов $p_1, p_2, p_3, \dots, p_M$; средние времена обработки заявок узлами $v_1, v_2, v_3, \dots, v_M$; стоимости узлов соответствующих уровней $c_1, c_2, c_3, \dots, c_M$.

Оценка частных показателей качества

Надежность системы оценим по вероятности сохранения работоспособности, которая зависит от формулировки условия отказа (сохранения функционирования Y). Если система работоспособна, когда на i -м уровне системы исправно не менее чем d_i узлов, то надежность системы [6, 7] составляет

$$P(m_1, m_2, m_3, \dots, m_M) = \sum_{k_1=d_1}^{m_1} \sum_{k_2=d_2}^{m_2} \sum_{k_3=d_3}^{m_3} \dots \sum_{k_M=d_M}^{m_M} C_{m_1}^{k_1} C_{m_2}^{k_2} C_{m_3}^{k_3} \dots C_{m_M}^{k_M} \times \\ \times p_1^{k_1} p_2^{k_2} p_3^{k_3} \dots p_M^{k_M} (1-p_1)^{m_1-k_1} (1-p_2)^{m_2-k_2} (1-p_3)^{m_3-k_3} \dots (1-p_M)^{m_M-k_M}$$

где $m_1, m_2, m_3, \dots, m_M$ – число узлов на каждом уровне, $C_m^k = m!/(k!(m-k)!)$ – число сочетаний из m по k .

Если условие работоспособности заключается в исправности на каждом уровне хотя бы одного узла [8, 9], то $P(m_1, m_2, m_3, \dots, m_M) = \prod_{i=1}^M (1-(1-p_i)^{m_i})$.

Представим каждый узел системой массового обслуживания типа $M/M/1$. Считая, что каждая заявка последовательно обслуживается одним из исправных узлов на каждом уровне системы, среднее время пребывания заявок в системе равна

$$U(m_1, m_2, m_3, \dots, m_M) = \sum_{i=1}^M v_i / (1 - v_i \lambda / m_i).$$

Максимальная интенсивность входного потока заявок

$$\Lambda(m_1, m_2, m_3, \dots, m_M) = \min_i (m_i / v_i).$$

Функцию для вычисления надежности в Mathcad зададим как

$$P(N) := \begin{cases} r \leftarrow 1 \\ \text{for } j \in 0..last(p) \\ r \leftarrow r \sum_{i=elem_min_j}^{N_j} combin(N_j, i) (p_j)^i (1-p_j)^{N_j-i} \\ r \end{cases}$$

Анализ приведенных целевых функций показывает необходимость обеспечения целочисленного поиска экстремумов и возможности поиска решений при нахождении искомого параметра оптимизации над знаком суммы в целевой функции, что не поддерживается встроенными средствами системы Mathcad.

Метод STEM

Метод STEM предусматривает выполнение многоэтапной процедуры [1]. Сначала проводится оптимизация по каждому критерию в отдельности, в результате формируется матрица значений критериев для найденных вариантов. Затем эта матрица нормализуется, и по ней находятся технические веса критериев. После этого проводится оптимизация по аддитивному критерию с использованием найденных весов. Результат предьявляется лицу, принимающему решение (ЛПР), которое может принять найденный вариант или изменить условия (ограничения) по частному критерию, не удовлетворяющему требованиям к системе, и этапы поиска решения повторяются.

Описание реализованных средств оптимизации

Система функций реализации метода STEM включает несколько вспомогательных и основных функций. Далее описаны только самые важные из них. Исходными данными для представленных функций является множество вариантов, заданное в виде матрицы, строки которой являются векторами критериев.

Вспомогательные функции нужны для выполнения некоторых операций с векторами. Функция $is_member(z, V)$ служит для проверки, является ли значение z элементом вектора V . Функция $fill_vector(value, length)$ возвращает вектор длины $length$, заполненный значением $value$. Функции $i_max(V)$ и $i_min(V)$ возвращают индексы максимального и минимального элемента в векторе.

Функция нормализации нормализует значения всех критериев в заданной матрице по столбцам и приводит их к виду «чем больше, тем лучше». Эта и другие функции принимают в качестве параметра вектор (L_b) («less better») с номерами критериев вида «чем меньше, тем лучше».

Функции для нахождения множества вариантов, удовлетворяющих заданным ограничениям. Функция $is_satisfied(Crits, Lim, L_b)$ проверяет значения критериев одного варианта на соответствие ограничениям, а функция $stem_R(R, Lim, L_b)$ возвращает матрицу из тех векторов критериев, которые удовлетворяют наложенным ограничениям:

```

stem_R(R, Lim, L_b) := | j ← 0
                       | for i ∈ 0..rows(R) - 1
                       |   if is_satisfied[(RT)(i), Lim, L_b]
                       |     | newR(j) ← (RT)(i)
                       |     | j ← j + 1
                       | newRT
    
```

Функция скалярной оптимизации,

```

stem_T(R, Lim, L_b) := | SR ← stem_R(R, Lim, L_b)
                       | Cols ← SRT
                       | for n ∈ 0..cols(SR) - 1
                       |   T(n) ← | Cols(i_max(SR(n)) if ¬is_member(n, L_b)
                       |         | Cols(i_max(SR(n)) otherwise
                       | TT,
    
```

находит варианты, лучшие по каждому из частных критериев (располагаемые по диагонали полученной матрицы).

Функции для нахождения технических весов критериев. По нормализованной матрице, формируемой функцией $stem_T(R, Lim, L_b)$ вычисляются средние значения элементов по столбцам (кроме диагонально расположенных элементов матрицы):

$$\alpha_j = \left[\left(\sum_{i=1}^m C_i^j \right) - C_j^j \right] / (m-1),$$

где m – количество критериев (количество строк в таблице).

Затем из соотношений $\frac{\lambda_i}{\lambda_j} = \frac{1 - \alpha_i}{1 - \alpha_j}$ и $\sum_{i=1}^m \lambda_i = 1$ находятся веса: $\lambda_i = (1 - \alpha_i) / \left(m - \sum_{j=1}^m \alpha_j \right)$.

Функции имеют вид

```

stem_αs(R, Lim, L_b) := | T ← normalize(stem_T(R, Lim, L_b), L_b)
                       | TT · fill_vector(1, cols(T)) - 1
                       | -----
                       | cols(T) - 1
stem_λs(R, Lim, L_b) := | αs ← stem_αs(R, Lim, L_b)
                       | -----
                       | length(αs) - ∑ αs
    
```

Функция оптимизации по глобальному критерию проводит оптимизацию по аддитивному критерию

$C_{gl} = \sum_{i=1}^m c_i \cdot \lambda_i$ с использованием формально найденных весов и возвращает вектор значений критериев оптимальной системы.

```

stem(R, Lim, L_b) := | SR ← stem(R, Lim, L_b)
                    | Cols ← normalize(SR, L_b)T
                    | λs ← stem_λs(R, Lim, L_b)
                    | max_value ← -∞
                    | for i ∈ 0..cols(Cols) - 1
                    |   | value ← λs · Cols(i)
                    |   | if value > max_value
                    |   |   | max_value ← value
                    |   |   | max_ind ← i
                    | (SRT)(max_ind)
    
```

Пример поиска решений

Рассмотрим применение предложенных функций при оптимизации кластера.

Зададим ограничения для критериев и вектор, указывающий, значения каких критериев при оптимизации надо максимизировать, а каких – минимизировать: $P0:=0,9$, $U0:=10$ с, $\Lambda0:=0$, $C0:=100$ у.е., вектор ограничений $L:=stack(P0,U0,\Lambda0,C0)$, вектор $L_b:=stack(1,3)$, это означает, что минимизируем U и C . Время пребывания будет рассчитываться при $\lambda:=0,2$ с.

Сформируем матрицу RN , содержащую значения критериев и имена систем, которые представляют собой перечисление числа узлов на каждом уровне в виде «a-b-c». Затем отделим критерии от имен и найдем область Парето, чтобы сузить поиск.

```
Pareto names filter
RN:= make_RN(L,L_b)
R:= RN_to_R(RN)
N:= RN_to_N(RN)
R_n:=normalize (R,L_b)
N_p:= pareto_names(R_n,N)
R_p:=filter_names(R,N,N_p)
```

R_p – матрица критериев для решений из области Парето. По этой матрице и будем проводить оптимизацию методом STEM.

Решение получаем, применяя функцию $stem(R_p,L,L_b)$, при этом искомый вектор критериев оптимальной системы с учетом заданных ограничений имеет вид

$$\begin{pmatrix} 0,999468 \\ 9,065934 \\ 1,5 \\ 51 \end{pmatrix}.$$

Предложенные средства позволяют отслеживать промежуточные шаги процесса поиска решения.

Первый шаг – оптимизация по частным критериям: $T:=stem_T(R_p,L,L_b)$.

Для заданных исходных данных результат получаем в виде матрицы

$$T = \begin{pmatrix} P & T & \Lambda & C \\ 1 & 8,554331 & 2,25 & 100 \\ 0,999997 & 8,500444 & 2,5 & 100 \\ 0,999999 & 8,571429 & 3 & 99 \\ 0,983566 & 10 & 1 & 33 \end{pmatrix}.$$

Второй шаг – нахождение технических весов, используя функцию $stem_\lambda s(R_p,L,L_b)$, в результате применения которой для рассматриваемого примера получаем вектор

$$\begin{pmatrix} 0,149437 \\ 0,161837 \\ 0,242768 \\ 0,445958 \end{pmatrix}.$$

Третий шаг – оптимизация по глобальному критерию. Вспомогательная функция позволяет получить имя и параметры выбранной системы:

```
characteristics:= stem(R_p,L,L_b)
name:= crits_to_name(characteristics,R_p,N_p)
name = "3-6-3"
```

В результате применения этой функции для рассматриваемого примера получаем вектор

$$characteristics = \begin{pmatrix} 0,999468 \\ 9,065934 \\ 1,5 \\ 51 \end{pmatrix}.$$

Четвертый шаг – уточнение условий. Результат предьявляется ЛППР, которое может указать, по какому частному критерию его не устраивает решение, после чего на соответствующий критерий установ-

ливается ограничение. Допустим, ЛПР не удовлетворено средним временем пребывания заявок и требует, чтобы оно было меньше 9 с. Из результатов оптимизации по частным критериям видно, что есть возможность снижать это время почти до 8,5 с. Изменяя ограничение для времени пребывания $U0:=9$ с, получим:

$$name = "4-8-4",$$

$$characteristics = \begin{pmatrix} 0,999959 \\ 8,77193 \\ 2 \\ 68 \end{pmatrix}.$$

Заключение

Показано, что решение задачи векторной оптимизации многоуровневых компьютерных систем кластерной архитектуры требует расширения возможностей системы Mathcad средствами целочисленного поиска экстремумов и поиска решений для целевой функции, содержащей искомые параметры оптимизации над знаком суммы.

Предложено расширение возможностей системы компьютерной математики Mathcad при создании набора средств (функций) поддержки поиска решений задач векторной оптимизации на основе метода STEM.

Литература

1. Ларичев О.И. Теория и методы принятия решений, а также Хроника событий в Волшебных странах: Учебник. – М.: Логос, 2002. – 392 с.
2. Воскобойников Ю.Е., Очков В.Ф. Программирование и решение задач в пакете Mathcad. – Новосибирск: НГАСУ, 2002. – 69 с.
3. Богатырев В.А., Булыгин К.А., Пинкевич В.Ю. Векторная оптимизация с программированием в среде Mathcad при комплексировании машин и агрегатов // Техничко-технологические проблемы сервиса. – 2011. – № 4. – С. 48–54.
4. Богатырев В.А., Богатырев С.В. Критерии оптимальности многоустойчивых отказоустойчивых компьютерных систем // Научно-технический вестник СПбГУ ИТМО. – 2009. – № 5 (63). – С. 92–98.
5. Богатырев В.А. Оценка надежности и оптимальное резервирование кластерных компьютерных систем // Приборы и системы. Управление, контроль, диагностика. – 2006. – № 10. – С. 18–21.
6. Богатырев В.А. Оптимальное резервирование системы разнородных серверов // Приборы и системы. Управление, контроль, диагностика. – 2007. – № 12. – С. 30–36.
7. Богатырев В.А., Богатырев С.В. К анализу и оптимизации серверных систем кластерной архитектуры с балансировкой нагрузки // Приборы и системы. Управление, контроль, диагностика. – 2010. – № 2. – С. 4–9.
8. Богатырев В.А., Богатырев С.В., Богатырев А.В. Оптимизация кластера с ограниченной доступностью кластерных групп // Научно-технический вестник СПбГУ ИТМО. – 2011. – № 1 (71). – С. 63–66.
9. Богатырев В.А., Богатырев С.В. Объединение резервированных серверов в кластеры высоконадежной компьютерной системы // Информационные технологии. – 2009. – № 6. – С. 41–47.

- | | |
|---------------------------------------|---|
| Булыгин Кирилл Александрович | – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, студент, kirill.bulygin@gmail.com |
| Пинкевич Василий Юрьевич | – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, студент, vasiliy.pinkevich@yandex.ru |
| Богатырев Владимир Анатольевич | – Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, доктор технических наук, профессор, vladimir.bogatyrev@gmail.com |