Формальная постановка задачи (ФПЗ): Матрицы расписания для независимых работ на однородных процессорах

(вариант K_2)

Мозжухин Александр, 421 группа

1 Дано

- Натуральные числа: число работ $N \in \mathbb{N}$, число процессоров $M \in \mathbb{N}$.
- Вектор длительностей работ $\mathbf{p} = (p_1, \ldots, p_N)$, где для всех $i \in \{1, \ldots, N\}$ выполняется $p_i \in \mathbb{N}$, $p_i \geq 1$.

Допущения: работы независимы; процессоры однородны; прерывания выполнения не допускаются; на одном процессоре в каждый момент времени выполняется не более одной работы.

2 Представление решения

Решение кодируется *матрицей расписания*

$$R \in \{0, 1, \ldots, N\}^{M \times N}.$$

Строка m $(1 \leq m \leq M)$ соответствует одному процессору. Ненулевые элементы строки образуют *упорядоченную* последовательность работ, выполняемых последовательно слева направо; нули — «пустые ячейки» (заполнитель).

Формальное определение строки. Для каждой строки m существует единственное $\ell_m \in \{0, \ldots, N\}$ такое, что

$$\underbrace{R_{m,1}, \ldots, R_{m,\ell_m}}_{\in\{1,\ldots,N\}}, \qquad R_{m,\ell_m+1} = \cdots = R_{m,N} = 0.$$

Обозначим $\pi_m = (R_{m,1}, \ldots, R_{m,\ell_m})$.

3 Условия корректности матрицы расписания

Матрица R корректна тогда и только тогда, когда выполнены условия **(C1)**–**(C3)**:

(C1) Диапазон значений. $\forall m \in \{1, \ldots, M\}$, $\forall j \in \{1, \ldots, N\}$: $R_{m,j} \in \{0, 1, \ldots, N\}$.

(C2) Единственность вхождения. Каждая работа встречается ровно один раз:

$$\forall i \in \{1, \ldots, N\}: \quad \big|\{(m, j) \mid R_{m,j} = i\}\big| = 1.$$

Эквивалентно: число ненулевых элементов в R равно N.

(C3) Контактность ненулей в строке. В каждой строке сначала идут все ненули, затем только нули (нулей «между» работами быть не должно), т.е.

$$\forall m,\ \forall j \in \{1, \ldots, N-1\}: \quad \mathbf{1}\{R_{m,j} \neq 0\} \geq \mathbf{1}\{R_{m,j+1} \neq 0\}.$$

Агрегированные ограничения назначения (форма с h_{im})

Введём бинарные переменные

$$h_{im} \in \{0, 1\}, \qquad h_{im} = 1 \iff \text{работа } i \text{ назначена процессору } m.$$

Для матрицы R положим $h_{im}(R) = \sum_{j=1}^{N} \mathbf{1}\{R_{m,j} = i\} \in \{0, 1\}$. Тогда агрегированная система ограничений записывается как

$$\boxed{\begin{cases} \displaystyle\sum_{i=1}^{N} \sum_{m=1}^{M} h_{im} = N, \\ \forall i \in \{1, \ldots, N\}: \quad \displaystyle\sum_{m=1}^{M} h_{im} = 1. \end{cases}} \tag{$\mathcal{H}$}$$

Она эквивалентна (C2); первая строка следует из второй при целочисленности h_{im}, но сохраняется в явном виде для соответствия требуемому формату.

4 Времена стартов и завершений

Пусть $R_{m_i,k_i} = i$. Тогда старт и завершение работы i определяются как

$$S_i(R) = \sum_{t=1}^{k_i - 1} p_{R_{m_i,t}}, \qquad C_i(R) = S_i(R) + p_i.$$

Завершение процессора m:

$$T_m(R) = \sum_{t=1}^{\ell_m} p_{R_{m,t}}.$$

Очевидно, $\max_i C_i(R) = \max_m T_m(R)$, а также $\min_i C_i(R)$ — минимум по завершениям отдельных работ.

5 Функция времени выполнения расписания (аналитический вид)

Введём кронекерову дельту $\delta(a, b)$ (равна 1 при $a = b$ и 0 иначе) и индикатор непустой ячейки $\mathbf{1}\{x \neq 0\}$.

Индексная форма (через матрицу R)

Время работы m-го процессора:

$$T_m(R, \mathbf{p}) = \sum_{j=1}^{N} p_{R_{m,j}} \cdot \mathbf{1}\{R_{m,j} \neq 0\}.$$

Время выполнения расписания (длина расписания, makespan):

$$T(R, \mathbf{p}) = \max_{m=1,\dots,M} T_m(R, \mathbf{p}) = \max_{i=1,\dots,N} C_i(R, \mathbf{p}).$$

Здесь $C_i(R, \mathbf{p})$ равны

$$C_i(R, \mathbf{p}) = \sum_{m=1}^{M} \sum_{j=1}^{N} \delta(R_{m,j}, i) \left(\sum_{t=1}^{j} p_{R_{m,t}} \right),$$

а стартовые времена:

$$S_i(R, \mathbf{p}) = \sum_{m=1}^{M} \sum_{j=1}^{N} \delta(R_{m,j}, i) \left(\sum_{t=1}^{j-1} p_{R_{m,t}} \right).$$

Благодаря (C3) «контактности» формулы корректно интерпретируют нули как хвостовые пустоты.

Линейная форма через бинарные переменные присвоения

Определим бинарные переменные

$$x_{mji} \in \{0, 1\}, \qquad x_{mji} = 1 \iff R_{m,j} = i.$$

Тогда

$$T_m(\mathbf{x}, \mathbf{p}) = \sum_{j=1}^{N} \sum_{i=1}^{N} p_i \, x_{mji}, \qquad T(\mathbf{x}, \mathbf{p}) = \max_{m=1,\dots,M} T_m(\mathbf{x}, \mathbf{p}).$$

Времена завершения работ:

$$C_i(\mathbf{x}, \mathbf{p}) = \sum_{m=1}^{M} \sum_{j=1}^{N} x_{mji} \left(\sum_{t=1}^{j} \sum_{i'=1}^{N} p_{i'} \, x_{mti'} \right).$$

Для эквивалентности с матричной моделью достаточно ограничений:

$$\begin{aligned}
&\forall i: \quad \sum_{m=1}^{M} \sum_{j=1}^{N} x_{mji} = 1 \quad \text{(каждая работа ровно один раз)},\\
&\forall m, j: \quad \sum_{i=1}^{N} x_{mji} \in \{0, 1\} \quad \text{(в ячейке либо одна работа, либо пусто)},\\
&\forall m, j < N: \quad \sum_{i} x_{mji} \;\geq\; \sum_{i} x_{m,j+1,i} \quad \text{(контактность)}.
\end{aligned}$$

Тогда «плотные» строки обеспечены, а $T(\mathbf{x}, \mathbf{p})$ является аналитической функцией времени выполнения.

6 Целевая функция (только K_2 в данном варианте)

Целевая функция — *сумма времён завершения работ*:

$$K_2(R) = \sum_{i=1}^{N} C_i(R).$$

Удобные эквивалентные формы:

$$\begin{aligned} K_2(R) &= \sum_{m=1}^{M}\sum_{j=1}^{N} \mathbf{1}\{R_{m,j} \neq 0\}\left(\sum_{t=1}^{j} p_{R_{m,t}}\right), \\ &= \sum_{m=1}^{M}\sum_{j=1}^{\ell_m} (\ell_m - j + 1)\, p_{R_{m,j}} \quad \text{(вклад строки как сумма «взвешенных» длительностей).} \end{aligned}$$

7 Требуется (формулировка задачи оптимизации)

$$\boxed{R^\star \in \arg\min_{R \in \{0,1,\dots,N\}^{M\times N}} K_2(R) \quad \text{при} \quad (C1) \wedge (C2) \wedge (C3) \quad \text{(и эквивалентно — при системе } (\mathcal{H})\text{).}}$$

8 Проверка ограничений после каждой итерации алгоритма (имитация отжига)

Пусть итерационный метод порождает кандидат $R^{(k)}$. Перед его приёмкой выполняется `is_valid`:

1. **Инициализация:** массив $cnt[1..N] \leftarrow 0$.
2. **Построчная проверка (C1)+(C3):** для каждой строки m:
   (a) $seenZero \leftarrow$ false;
   (b) для $j = 1..N$:
      - (C1) если $R_{m,j} \notin \{0, 1, \dots, N\}$: `return false`;
      - если $R_{m,j} = 0$: $seenZero \leftarrow$ true;
      - если $R_{m,j} \neq 0$ и $seenZero =$ true: нарушено (C3) $\Rightarrow$ `return false`;
      - если $R_{m,j} \neq 0$: увеличить $cnt[R_{m,j}]$ на 1.
3. **Единственность (C2):** если существует i с $cnt[i] \neq 1$, то `return false`.
4. **Успех:** `return true`.

При `true` вычисляем $K_2(R^{(k)})$ по формулам выше; при `false` шаг отвергается.

Замечание (проверка системы $(\mathcal{H})$). Альтернативно можно вычислить $h_{im}(R^{(k)}) = \sum_j \mathbf{1}\{R^{(k)}_{m,j} = i\}$ и проверить $\forall i : \sum_m h_{im} = 1$ и $\sum_{i,m} h_{im} = N$. Это эквивалентно (C2) и также реализуется за $O(MN)$.

9 Совместимость с имитацией отжига (кратко)

Состояние — матрица R. Соседство $\mathcal{N}(R)$ задаём так, чтобы автоматически сохранялись (C1)–(C3): перемещение/вставка работы между позициями (и/или процессорами), swap двух работ в строке, 2-opt в строке. Правило приёмки: улучшения ($\Delta K_2 \le 0$) принимаются всегда, ухудшения — с вероятностью $\exp(-\Delta K_2/T)$. Охлаждение — по выбранному закону (геометрический, линейный, Коши и др.). Остановка — по отсутствию улучшения в течение заданного числа шагов либо по порогу температуры.

10 Примеры

Пример 1 (корректная матрица и вычисление K_2)

$N = 5$, $M = 2$, $\mathbf{p} = (3, 2, 4, 1, 2)$.

$$R = \begin{pmatrix} 1 & 3 & 0 & 0 & 0 \\ 2 & 4 & 5 & 0 & 0 \end{pmatrix}.$$

Проверки: (C1)–(C3) выполнены. Вклады строк:

$$\ell_1 = 2: \quad (2-1+1)p_1 + (2-2+1)p_3 = 2 \cdot 3 + 1 \cdot 4 = 10;$$

$$\ell_2 = 3: \quad (3-1+1)p_2 + (3-2+1)p_4 + (3-3+1)p_5 = 3 \cdot 2 + 2 \cdot 1 + 1 \cdot 2 = 10.$$

Итого $K_2(R) = 20$. Для справки: $T_1 = 7$, $T_2 = 5$, $T(R) = 7$.

Пример 2 (нарушение контактности, (C3))

$N = 4$, $M = 1$, $\mathbf{p} = (1, 1, 1, 1)$,

$$R = \begin{pmatrix} 1 & 0 & 3 & 0 \end{pmatrix} \quad \Rightarrow \quad \text{после первого нуля встречается ненуль 3} \;\Rightarrow\; \text{нарушено (C3).}$$

Пример 3 (нарушение единственности, (C2))

$N = 3$, $M = 2$, $\mathbf{p} = (2, 2, 2)$,

$$R = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 3 & 0 \end{pmatrix} \quad \Rightarrow \quad \text{работа 2 встречается дважды} \;\Rightarrow\; \text{нарушено (C2).}$$

11 Спецификация вычислений и проверок

Предусловия. $M, N \in \mathbb{N}$, $p_i \in \mathbb{N}$, $p_i \ge 1$, $R \in \{0, \dots, N\}^{M \times N}$.

Постусловия (корректность). Возвращаемое расписание R удовлетворяет (C1)–(C3) (и эквивалентно — системе $(\mathcal{H})$).

Вычисление S_i, C_i, T_m, K_2.

- Один линейный проход по каждой строке с аккумулятором префиксной суммы $t \leftarrow 0$: для каждого ненулевого $R_{m,j}$: положить $S_{R_{m,j}} \leftarrow t$, затем $t \leftarrow t + p_{R_{m,j}}$, после чего $C_{R_{m,j}} \leftarrow t$. По завершении строки $T_m \leftarrow t$.
- $T(R, \mathbf{p}) = \max_m T_m$; $\quad K_2(R) = \sum_{i=1}^{N} C_i$.

Сложность: $O(MN)$ времени и $O(N)$ памяти.

12 Интерфейсы (рекомендации для реализации на C++)

- Проверка корректности:

```
bool is_valid(const Matrix& R, const std::vector<int>& p);
```

- Оценка расписания (целевой функции K_2):

```
long long objective_K2(const Matrix& R, const std::vector<int>& p);
```

- (Необязательно, для анализа) Время выполнения расписания:

```
long long makespan(const Matrix& R, const std::vector<int>& p);
```