СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Ивницкий, В.А. Теория сетей массового обслуживания / В.А. Ивницкий. – М.: Физико-математическая литература, 2004. – 772 с.

2. Муравьев, Г.Л. Компьютерная генерация спецификаций сетевых архитектур заданной сложности / Г.Л. Муравьев, А.Н. Никонюк, В.И. Хвещук // Технологии информатизации и управления: сб. 2-й Межд. научно-практич. конф. (TIM-2011) – Минск: БГУ, 2011 – С. 50 – 53.

3. Рыжиков, Ю.И. Имитационное моделирование. Теория и технологии / Ю.И. Рыжиков. – СПб.: Корона, 2004. – 320 с.

УДК 519.853.3

Ракитский А.В.

Научный руководитель: доцент Ракецкий В.М.

РЕШЕНИЕ ПРОСТОЙ ЗАДАЧИ КВАДРАТИЧНОГО ПРОГРАММИРОВАНИЯ ПРЯМЫМ ОПОРНЫМ МЕТОДОМ

1. Постановка задачи. Рассмотрим задачу выпуклого программирования

$$F(x) = \frac{1}{2}x'Dx + c'x \to min \qquad (1)$$

при простых ограничениях

$$d_* \le x \le d^*, \qquad (2)$$

где $x = x(J)$ – n-вектор неизвестных, $D = D(J,J) = \{d_{ij}, i, j \in J\} \ge 0$, – постоянная матрица, $d_* = d_*(J)$, $d^* = d^*(J)$ – соответственно нижняя и верхняя границы на значения переменных, $J = \{1,2, ..., n\}$. Для решения задачи (1) (без ограничений) успешно используются различные методы сопряженных направлений. Однако наличие простых ограничений (2) существенно снижает эффективность этих методов, так как при каждом выходе на границу области допустимых точек процедуру построения сопряженных направлений приходится начинать заново.

Прямой метод [1-3] естественным образом учитывает структуру ограничений (2) и не требует «обнуления» итерационной процедуры при выходе на границу допустимой области.

2. Алгоритм прямого опорного метода. Допустим, что $\{x, J_{оц}\}$ – согласованный опорный план (СОП) задачи (1) [1-3], $H_{оц} = D(J_{оц}, J_{оц})$, $G_{оц} = H_{оц}^{-1}$. Каждая итерация прямого метода состоит из трех шагов.

Шаг 1. Проверка достаточных условий оптимальности. Вычислим вектор оценок (градиент) целевой функции

$$\Delta = Dx + c \qquad (3)$$

и проверим соотношения:

$$\Delta_j \ge 0 \text{ при } x_j = d_{*j};\ \Delta_j \le 0 \text{ при } x_j = d_j^*;\ \Delta_j = 0 \text{ при } d_{*j} < x_j < d_j^*,\ j \in J_н = J \setminus J_{оц}. \qquad (4)$$

Если условия (4) выполняются, то решение задачи (1), (2) окончено, $\{x, J_{оц}\}$ – оптимальный опорный план.

Допустим, условия (4) не выполняются. В этом случае переходим к шагу 2.

Шаг 2. Построение направления улучшения СОП $\{x, J_{оц}\}$ и расчет максимально допустимого плана. Среди оценок $\Delta_j, j \in J_н$, не удовлетворяющих условиям оптимальности, выберем оценку Δ_{j_0} с максимальной по модулю величиной. Подходящее для СОП $\{x, J_{оц}\}$ направление построим по формулам:

$$l_{j_0} = -sign\Delta_{j_0},\quad l_j = 0,\quad j \in J_н \setminus j_0,\ l_{оц} = G_{оц}\, p_{оц}\, sign\Delta_{j_0} \text{ при } J_{оц} \ne \varnothing, \qquad (5)$$

где $p_{оц} = D(J_{оц}, j_0)$.

Вычислим максимально допустимый шаг θ^0 вдоль направления l:

$$\theta^0 = \min\{\theta_{оп}, \theta_\alpha, \theta_{j_0}\}, \qquad (6)$$

где $\theta_{оп}$ – ограничение на шаг, вытекающее из учета ограничений по опорным компонентам:

$$\theta_{оп} = \begin{cases} \infty, \text{ если } J_{оп} = \varnothing, \\ \min\limits_{j \in J_{оп}} \theta_j, \text{ если } J_{оп} \neq \varnothing, \end{cases} \quad \theta_j = \begin{cases} (d_{*j} - x_j)/l_j, \text{ если } l_j < 0, \\ (d_j^* - x_j)/l_j, \text{ если } l_j > 0, \\ \infty, \text{ если } l_j = 0. \end{cases}$$

θ_α – ограничение на шаг, вытекающее из условия минимума целевой функции вдоль направления $F(x+\theta_\alpha l) = \min\limits_\theta F(x+\theta l)$:

$$\theta_\alpha = \begin{cases} |\Delta_{j_0}|/\alpha, \text{ если } \alpha > 0, \\ \infty, \text{ если } \alpha = 0, \end{cases} \quad \alpha = \begin{cases} d_{j_0 j_0}, \text{ если } J_{оп} = \varnothing, \\ p_{оп} sign\,\Delta_{j_0} + d_{j_0 j_0}, \text{ если } J_{оп} \neq \varnothing, \end{cases}$$

θ_{j_0} – ограничение на шаг, вытекающее из учета ограничений по компоненте x_{j_0}.

Шаг 3. Построение нового СОП. Вычислим новый план $\bar{x} = x + \theta^0 l$. В зависимости от результатов вычисления максимально допустимого шага построим новую опору $\tilde{J}_{оп}$ и соответствующую ей матрицу $\tilde{G} = \left[D\left(\tilde{J}_{оп}, \tilde{J}_{оп}\right)\right]^{-1}$:

а) $\theta^0 = \theta_{j_0}$. Полагаем $\tilde{J}_{оп} = J_{оп}$ и переходим к следующей итерации. Поскольку опора целевой функции не изменилась, то $\tilde{G}_{оп} = G_{оп}$ (при $J_{оп} \neq \varnothing$).

б) $\theta^0 = \theta_\alpha$. Полагаем $\tilde{J}_{оп} = J_{оп} \cup j_0$. Если $J_{оп} = \varnothing$, то

$$\tilde{G}_{оп} = \left\{\frac{1}{d_{j_0 j_0}}\right\}; \qquad (10)$$

в противном случае $\tilde{G}_{оп}$ вычисляется по формуле [4]

$$\tilde{G}_{оп} = \begin{pmatrix} G_{оп} + \dfrac{l_{оп} l'_{оп}}{\alpha} & -\dfrac{l_{оп}}{\alpha} sign\Delta_{j_0} \\ -\dfrac{l'_{оп}}{\alpha} sign\Delta_{j_0} & \dfrac{1}{\alpha} \end{pmatrix}.$$

в) $\theta^0 = \theta_{оп} = \theta_{i_0}$. Полагаем $\tilde{J}_{оп} = J_{оп} \setminus j_0$. При этом если $\tilde{J}_{оп} \neq \varnothing$, то

$$\tilde{G}_{оп} = G(\tilde{J}_{оп}, \tilde{J}_{оп}) - \frac{G(i_0, \tilde{J}_{оп}) G(\tilde{J}_{оп}, i_0)}{G(i_0, i_0)}.$$

Итерационное применение шагов 1-3 к задаче (1),(2) позволяет за конечное число итераций найти ее оптимальный план.

3. Описание модификации прямого опорного метода. В алгоритме, описанном в п.2, весьма трудоемкой является процедура проверки достаточных условий оптимальности, поскольку требуется вычисление градиента. Ведущий индекс находится по правилу:

$$|\Delta_{j_0}| = \max_{j \in J_н^-} |\Delta_j|,$$

где $J_н^-$ – множество индексов, на которых не выполняются достаточные условия оптимальности (4). Однако пара $\{x, J_{оп}\}$ по определению является согласованной, т.е.

$$\Delta(J_{оп}) = 0,$$

что позволяет не вычислять опорные компоненты градиента.

Изменим правило выбора ведущего индекса: вместо максимальной по модулю компоненты Δ_{j_0}, $j \in J_н^-$ будем выбирать первую обнаруженную не удовлетворяющую условиям оптимальности (4) компоненту Δ_j, $j \in J_н$. Такое изменение позволяет существенно облегчить процедуру проверки условий оптимальности, т.к. градиент будет вычисляться только частично.

Оценить теоретически влияние внесенных изменений на эффективность метода невозможно. Поэтому проведен численный эксперимент, в ходе которого проведено сравнение базовой версии алгоритма и описанной выше модификации.

4. Результаты численного эксперимента. Базовая версия алгоритма и его модификация реализованы программно на языке программирования С++. Проведен численный эксперимент, результаты которого приведены в таблице 1:

Таблица 1 – Результаты численного эксперимента

m	*n*	Базовый		Модифицированный		Небазисные элементы, %	Эффект., %
		Время	Итераций	Время	Итераций		
5	50	16026	91	3079	142	90	520,49
20	50	18417	92	8818	133	60	208,86
50	50	13492	46	7699	49	13	175,24
100	100	144844	87	68055	98	14	212,83

Здесь *m* – ранг матрицы *D*, *n* – количество переменных, время – количество тактов процессора.

Анализ результатов показал, что модифицированная версия алгоритма в 1,5-5 раз эффективнее базовой версии.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Габасов, Р. К методам решения общей задачи выпуклого квадратичного программирования / Р. Габасов, Ф. М. Кириллова, В.М. Ракецкий: докл. АН СССР. – 1981. – Т.258, N6. – С. 1289-1293.

2. Ракецкий, В.М. Прямой опорный метод квадратичного программирования: Проблемы оптимального управления. – Минск: Наука и техника, 1981. – С. 318 – 335.

3. Габасов, Р. Конструктивные методы оптимизации / Р. Габасов, Ф.М. Кириллова, О.И. Костюкова, В.М. Ракецкий. – Мн.: Изд-во "Университетское", 1987. – Ч.4: Выпуклые задачи. – 223 с.

4. Фаддеев, Д.К. Вычислительные методы линейной алгебры / Д.К. Фаддеев, В.Н. Фаддеева. – М.: Физматгиз,1960. – 734 с.

УДК 681.3

Рыщук А.С.

Научный руководитель: проф. Муравьев Г.Л.

ТРУДОЁМКОСТЬ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ СТОХАСТИЧЕСКИХ СЕТЕЙ

Наряду с адекватностью, результативностью и другими характеристиками степень полезности и применимости моделей и в особенности имитационных существенно зависит от их сложности, трудоемкости. Характеристики моделей определяются в процессе их аттестации, используются для оценки собственных свойств моделей, анализа влияющих факторов, для сравнения с характеристиками аналогичных моделей [1, 2].

Для стохастических сетевых моделей (ССМ) [3] сложность определяется особенностями архитектуры модели – сложностью структуры сети узлов и потоков заявок, сложностью управления обслуживанием и может быть оценена по количеству и составу параметров модели, необходимых для ее спецификации.

Трудоемкость модели как сложность вычислений оценивается количеством вычислительной работы (времени, операций, команд и т.д.), требуемой для ее реализации с целью получения набора характеристик заданной полноты и точности. Трудоемкость модели связана со сложностью самой сети. Но при ее определении учитываются особен-