

3. Решение систем уравнений и неравенств алгоритмами внутренних точек

Решение систем линейных уравнений и неравенств с интервальными ограничениями на переменные является одной из областей, где алгоритмы внутренних точек могут особенно успешно применяться на практике. Это справедливо как для алгоритмов аффинно-масштабирующего метода, уже зарекомендовавшего себя на практических задачах, так и для рассмотренных в предыдущей главе алгоритмов центрального и скошенного пути.

Отдельные вопросы, касающиеся систем линейных уравнений и неравенств, возникали еще в XIX веке, однако активные исследования в этой области можно целиком отнести к двадцатому столетию. Основные результаты теории конечных систем линейных уравнений и неравенств, включая алгебраические методы их решения, изложены С.Н.Черниковым в [35].

На практике системы линейных уравнений и неравенств чаще всего решаются на основе их сведения к задачам линейного программирования. Идеи такого перехода затрагивались в первой главе диссертационной работы. Среди последних работ в данной области можно отметить [4].

3.1. Задача определения допустимых режимов электроэнергетических систем

Применение алгоритмов для решения нелинейных систем

Необходимость решения систем линейных уравнений и неравенств часто возникает в связи с реализацией нелинейных моделей из различных прикладных областей на основе методов последовательной линеаризации. Многие такие модели сводятся к решению относительно вектора \mathbf{z} системы следующего вида:

$$\mathbf{W}(\mathbf{z}) = \mathbf{0}, \quad \bar{\mathbf{z}} \geq \mathbf{z} \geq \underline{\mathbf{z}}. \quad (1)$$

Здесь $\bar{\mathbf{z}}$ и $\underline{\mathbf{z}}$ - заданные векторы ограничений, $\mathbf{W}(\mathbf{z})$ - нелинейная дифференцируемая вектор-функция.

В частности, в виде (1) представляется задача поиска допустимых режимов электроэнергетических систем (ЭЭС) [25], [26], [31], нерешенные аспекты которой и инициировали представляемые в диссертационной работе исследования. Данная задача является частным примером, однако основные полученные для нее выкладки могут быть полезны и для решения других систем нелинейных уравнений и неравенств.

Соотношения, раскрывающие содержание условий задачи определения допустимых режимов ЭЭС

Ограничения-равенства:

В соответствии с законами Кирхгофа для каждого i -го узла сети выполняются следующие соотношения

$$\sum_{j \neq i} P_{ij} + P_{Hi} - P_{Gi} = 0, \quad \sum_{j \neq i} Q_{ij} + Q_{Hi} - Q_{Gi} = 0, \quad i = 1, \dots, n,$$

где P_{ij}, Q_{ij} - перетоки активной и реактивной мощности в ветви, соединяющей узлы i и j ; P_{Hi}, Q_{Hi} - мощности активной и реактивной нагрузки в i -ом узле, значения которых принимаются постоянными; P_{Gi}, Q_{Gi} - мощности активной и реактивной генерации в i -ом узле.

Соотношения для определения значений перетоков активной и реактивной мощности в ветви $(i - j)$ содержат комплексные величины напряжения в i -узле (U_{ia}, U_{ir}) , комплексные коэффициенты трансформации (K_{ija}, K_{ijr}) и (K_{jia}, K_{jir}) , включенные соответственно у i -го и j -го конца ветви; продольные проводимости ветви (y_{ija}, y_{ijr}) и поперечные проводимости у i -го и j -го конца ветви (y_{iija}, y_{iijr}) и (y_{jjja}, y_{jjjr}) :

$$\begin{aligned}
P_{ij} &= K_{ij}^2 U_i^2 (y_{ija} + y_{ijr}) - \\
&- (K_{ija} K_{jia} + K_{ijr} K_{jir}) ((U_{ia} U_{ja} + U_{ir} U_{jr}) y_{ija} - (U_{ia} U_{jr} - U_{ir} U_{ja}) y_{ijr}) + \\
&+ (K_{ija} K_{jir} - K_{ijr} K_{jia}) ((U_{ia} U_{jr} - U_{ir} U_{ja}) y_{ija} + (U_{ia} U_{ja} + U_{ir} U_{jr}) y_{ijr}), \\
Q_{ij} &= K_{ij}^2 U_i^2 (y_{ijr} + y_{ijr}) - \\
&- (K_{ija} K_{jia} + K_{ijr} K_{jir}) ((U_{ia} U_{jr} - U_{ir} U_{ja}) y_{ija} + (U_{ia} U_{ja} + U_{ir} U_{jr}) y_{ijr}) - \\
&- (K_{ija} K_{jir} - K_{ijr} K_{jia}) ((U_{ia} U_{ja} + U_{ir} U_{jr}) y_{ija} - (U_{ia} U_{jr} - U_{ir} U_{ja}) y_{ijr}).
\end{aligned}$$

В приведенных формулах U_i - модуль напряжения в i -узле, K_{ij} - модуль коэффициента трансформации, включенного у i -го конца ветви ($i - j$).

Ограничения-неравенства:

Для всех узлов должны выполняться условия

$$\bar{U}_i \geq U_i \geq \underline{U}_i, \quad \bar{P}_{Gi} \geq P_{Gi} \geq \underline{P}_{Gi}, \quad \bar{Q}_{Gi} \geq Q_{Gi} \geq \underline{Q}_{Gi},$$

а для всех ветвей - условия

$$\bar{P}_{ij} \geq P_{ij} \geq \underline{P}_{ij}, \quad \bar{K}_{ij} \geq K_{ij} \geq \underline{K}_{ij}.$$

Здесь $\bar{U}_i, \underline{U}_i, \bar{P}_{Gi}, \underline{P}_{Gi}, \bar{Q}_{Gi}, \underline{Q}_{Gi}, \bar{P}_{ij}, \underline{P}_{ij}, \bar{K}_{ij}, \underline{K}_{ij}$ - заданные максимальные и минимальные значения каждого из параметров режима.

Все приведенные параметры режима образуют вектор \mathbf{z} . Вектор \mathbf{z} разбивается на два вектора: независимых переменных \mathbf{x} (вектор регулируемых параметров) и зависимых переменных \mathbf{y} (базис уравнений). Разбиение производится таким образом, что при заданном \mathbf{x} система уравнений $\mathbf{W}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ разрешима относительно вектора \mathbf{y} , то есть можно получить неявную функцию $\mathbf{y}(\mathbf{x})$, такую что $\mathbf{W}(\mathbf{x}, \mathbf{y}(\mathbf{x})) \equiv \mathbf{0}$.

Существуют различные процедуры выбора состава компонент вектора \mathbf{x} , использование которых приводит соответственно к различным наборам регулируемых параметров. Не любой базис обеспечивает получение решения. В связи с этим логическим продолжением проблемы выбора базиса

является проблема смены базиса. Достаточно подробно эффективность применения различных методов выбора и смены базиса исследуется в [5].

Итогом разбиения переменных является переход к так называемой второй форме математического описания задачи

$$\mathbf{W}(\mathbf{x}, \mathbf{y}) = \mathbf{0}, \quad \mathbf{x} \geq \underline{\mathbf{x}} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{y} \geq \underline{\mathbf{y}}. \quad (2)$$

Итеративная процедура

Алгоритм решения задачи (2) основан на сочетании метода приведенного градиента и последовательной линеаризации. На каждой итерации $t = 0, 1, 2, \dots$ (эту итерацию назовем “глобальной”, чтобы отличать от “локальных” итераций, возникающих при решении линеаризованной задачи) выполняем [26] следующие действия:

1. Для заданного допустимого по ограничениям-неравенствам вектора $\mathbf{x}^t \in [\underline{\mathbf{x}}; \bar{\mathbf{x}}]$ находим вектор \mathbf{y}^t , удовлетворяющий условию $\mathbf{W}(\mathbf{x}^t, \mathbf{y}^t) = \mathbf{0}$.

2. Проверяем, удалось ли получить допустимый режим. Если справедливо условие $\mathbf{y}^t \in [\underline{\mathbf{y}}; \bar{\mathbf{y}}]$, то завершаем вычислительный процесс.

3. Определяем состав ограничений, учитываемых на данной итерации. Номера этих ограничений образуют множество \mathbf{I}_v^t . В него обязательно входят номера i , для которых компоненты y_i^t не удовлетворяют ограничениям-неравенствам: $y_i^t \notin [\underline{y}_i; \bar{y}_i]$. Также сюда могут входить номера i , для которых компоненты y_i^t в некотором смысле близки к границам. В частности, в одном из вариантов алгоритма в множество \mathbf{I}_v^t также включались номера i , такие что $y_i^t \notin [\underline{y}_i + \Delta_i; \bar{y}_i - \Delta_i]$, где Δ_i - заданные положительные величины, в другом - такие что $y_i^{t-1} \notin [\underline{y}_i; \bar{y}_i]$.

Данный пункт алгоритма совместно с выбором хорошего базиса позволяют существенно уменьшить размерность исходной задачи. В то же

время следует отметить, что рассматриваемая схема алгоритма может быть использована и при отсутствии исключения части ограничений.

4. Пусть $\mathbf{I}_v^t = \{1, \dots, m\}$. Далее будем считать, что вектор-функция \mathbf{W} состоит только из компонент с номерами из \mathbf{I}_v^t . Полагаем $\mathbf{B}^t = \nabla_y \mathbf{W}(\mathbf{x}^t, \mathbf{y}^t)$, $\mathbf{C}^t = \nabla_x \mathbf{W}(\mathbf{x}^t, \mathbf{y}^t)$. Линеаризуем условие $\mathbf{W}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$ в точке $(\mathbf{x}^t, \mathbf{y}^t)$:

$$\mathbf{W}(\mathbf{x}^t, \mathbf{y}^t) + \mathbf{B}^t(\mathbf{y} - \mathbf{y}^t) + \mathbf{C}^t(\mathbf{x} - \mathbf{x}^t) = \mathbf{0}.$$

Раскрывая скобки, получаем

$$\mathbf{y} - \hat{\mathbf{y}}^t = \mathbf{A}^t \mathbf{x},$$

где $\mathbf{A}^t = -(\mathbf{B}^t)^{-1} \mathbf{C}^t$, $\hat{\mathbf{y}}^t = \mathbf{y}^t + (\mathbf{B}^t)^{-1} \mathbf{C}^t \mathbf{x}^t$.

5. Решаем точно или приближенно задачу

$$\alpha \rightarrow \min, \quad \mathbf{y} = \mathbf{A}^t \mathbf{x} + \hat{\mathbf{y}}^t, \quad \bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{y} + \alpha \mathbf{r}^t \geq \underline{\mathbf{y}}, \quad \alpha \geq 0. \quad (3)$$

относительно переменных \mathbf{x} , \mathbf{y} и α . Здесь $r_i^t = (\bar{y}_i + \underline{y}_i)/2 - y_i^t$, $i = 1, \dots, m$.

Поскольку набор $(\mathbf{x}^t, \mathbf{y}^t, \alpha = 1)$ является допустимым, данная задача всегда имеет решение при некотором $\alpha \in [0; 1]$. Параметр α здесь является показателем степени несовместности линеаризованной системы. Если оптимальное значение α равняется нулю, то система

$$\bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{A}^t \mathbf{x} + \hat{\mathbf{y}}^t \geq \underline{\mathbf{y}}$$

совместна. Чем ближе оптимальное значение α к единице, тем больше будут нарушаться ограничения. Обозначим решение задачи (3) $\tilde{\mathbf{x}}^t$, $\tilde{\mathbf{y}}^t$.

6. Осуществляем итеративный переход по правилам:

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \rho_t (\tilde{\mathbf{x}}^t - \mathbf{x}^t), \quad \mathbf{y}^{t+1} = \mathbf{y}^t + \rho_t (\tilde{\mathbf{y}}^t - \mathbf{y}^t),$$

где шаг ρ_t выбирается, исходя из двух требований. Во-первых, необходимо, чтобы векторы \mathbf{x}^{t+1} и \mathbf{y}^{t+1} находились соответственно в интервалах $[\underline{\mathbf{x}}; \bar{\mathbf{x}}]$ и

$[\underline{y}; \bar{y}]$. Это достигается введением условия $\rho_t \leq 1$. Во-вторых, должна минимизироваться (точно или приближенно) функция

$$\phi(\rho) = \left\| \mathbf{W}(\mathbf{x}^t + \rho(\tilde{\mathbf{x}}^t - \mathbf{x}^t), \mathbf{y}^t + \rho(\tilde{\mathbf{y}}^t - \mathbf{y}^t)) \right\|.$$

Линеаризованная подзадача

Наиболее вычислительно емким в алгоритме, как видим, является пункт 5. В некоторых случаях не обязательно или даже нецелесообразно решать точно линеаризованную подзадачу (3). Может оказаться достаточным получить некое улучшающее решение, соответствующее некоторой величине α , являющейся верхней оценкой ее оптимального значения. То есть вариантом пункта 5 алгоритма может являться решение системы уравнений и неравенств

$$\mathbf{y} = \mathbf{A}^t \mathbf{x} + \tilde{\mathbf{y}}^t, \quad \bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{y} + \alpha \mathbf{r}^t \geq \underline{\mathbf{y}} \quad (4)$$

при задаваемом итеративно увеличиваемом $\alpha \in [0; 1]$. Для данной модификации алгоритма особенно важную роль играет максимально быстрая идентификация несовместности системы линейных уравнений и неравенств.

Рассмотрим систему линейных уравнений и неравенств относительно векторов переменных $\mathbf{x} \in \mathbf{R}^n$ и $\mathbf{y} \in \mathbf{R}^m$

$$\mathbf{A}\mathbf{x} - \mathbf{y} = \mathbf{0}, \quad \bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{y} \geq \underline{\mathbf{y}}. \quad (5)$$

Здесь $\mathbf{A} \in \mathbf{R}^{m \times n}$ - заданная матрица, $\bar{\mathbf{x}}, \underline{\mathbf{x}} \in \mathbf{R}^n$ и $\bar{\mathbf{y}}, \underline{\mathbf{y}} \in \mathbf{R}^m$ - заданные векторы ограничений, причем $\bar{\mathbf{x}} > \underline{\mathbf{x}}$, $\bar{\mathbf{y}} > \underline{\mathbf{y}}$. Необходимо либо отыскать допустимое решение системы (5), либо максимально быстро идентифицировать факт ее несовместности. Система (4) сводится к виду (5) путем замены переменных $\mathbf{y} := \mathbf{y} + \tilde{\mathbf{y}}^t$ и сдвига ограничений $\bar{\mathbf{y}} := \bar{\mathbf{y}} + \tilde{\mathbf{y}}^t - \alpha \mathbf{r}^t$, $\underline{\mathbf{y}} := \underline{\mathbf{y}} + \tilde{\mathbf{y}}^t - \alpha \mathbf{r}^t$.

3.2. Аффинно-масштабирующие алгоритмы применительно к решению линеаризованной подзадачи

Перейдем непосредственно к алгоритмам решения системы (5). Начнем рассмотрение с аффинно-масштабирующих алгоритмов, применяющихся с начала 80-х годов и хорошо показавших себя на практике. В частности, на задачах управления режимами они были реализованы в разработанном в ИСЭМ СО РАН программно-вычислительном комплексе СДО [26]. Тем не менее ряд усовершенствований позволяет еще в большей степени повысить эффективность вычислительного процесса.

Сведение к задаче линейного программирования

Пусть имеется некоторая стартовая точка $(\mathbf{x}^1, \mathbf{y}^1)$, удовлетворяющая в строгой форме ограничениям-неравенствам:

$$\bar{x}_j > x_j^1 > \underline{x}_j, \quad \bar{y}_i > y_i^1 > \underline{y}_i. \quad (6)$$

Стартовая точка может либо быть сформирована автоматически, например, по правилу

$$\mathbf{x}^1 = 0.5(\bar{\mathbf{x}} + \underline{\mathbf{x}}), \quad \mathbf{y}^1 = 0.5(\bar{\mathbf{y}} + \underline{\mathbf{y}}),$$

либо на основе априори имеющейся информации. В частности, если мы рассматриваем итеративный процесс решения нелинейной задачи, то приближение, полученное на предыдущей итерации линеаризации, может с учетом небольшой корректировки рассматриваться в качестве стартового на последующей.

Пусть $\mathbf{Ax}^1 \neq \mathbf{y}^1$, то есть стартовая точка не является допустимой для системы (5). Вычислим вектор невязок ограничений-равенств

$$\mathbf{r} = \mathbf{y}^1 - \mathbf{Ax}^1.$$

Сведение системы (5) к задаче линейного программирования происходит аналогично случаю, описанному в первой главе диссертационной работы. Если оптимальное решение α задачи

$$\alpha \rightarrow \min, \quad \mathbf{Ax} - \mathbf{y} + \alpha \mathbf{r} = \mathbf{0}, \quad \bar{\mathbf{x}} \geq \mathbf{x} \geq \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} \geq \mathbf{y} \geq \underline{\mathbf{y}}, \quad \alpha \geq 0 \quad (7)$$

равно нулю, то соответствующие векторы \mathbf{x} и \mathbf{y} являются допустимыми в (5), иначе система (5) несовместна.

Прямые алгоритмы

Приведем вычислительный процесс улучшения решения задачи (7). На каждой итерации $k = 1, 2, \dots$ выполняются следующие операции.

1. Вычисление вектора невязок

$$\mathbf{r}^k = \mathbf{y}^k - \mathbf{Ax}^k. \quad (8)$$

2. Нахождение диагональных матриц \mathbf{X}_k и \mathbf{Y}_k весовых коэффициентов, конкретизируемых в дальнейшем.

3. Поиск направления корректировки переменных с помощью решения вспомогательной задачи

$$(\Delta \mathbf{x})^T \mathbf{X}_k^{-1} \Delta \mathbf{x} + (\Delta \mathbf{y})^T \mathbf{Y}_k^{-1} \Delta \mathbf{y} \rightarrow \min_{\Delta \mathbf{x} \in \mathbf{R}^n, \Delta \mathbf{y} \in \mathbf{R}^m}, \quad \mathbf{A} \Delta \mathbf{x} - \Delta \mathbf{y} = \mathbf{r}^k. \quad (9)$$

4. Идентификация несовместности системы (5).

5. Поиск максимального шага $\tilde{\lambda}_k$, при движении с которым вдоль найденного направления корректировки решение остается внутри интервальных ограничений задачи (7). Для его нахождения вычислим величины

$$\lambda_k^1 = \min_{j: \Delta x_j^k < 0} \left\{ \frac{\underline{x}_j - x_j^k}{\Delta x_j^k} \right\}, \quad \lambda_k^2 = \min_{j: \Delta x_j^k > 0} \left\{ \frac{\bar{x}_j - x_j^k}{\Delta x_j^k} \right\},$$

$$\lambda_k^3 = \min_{i: \Delta y_i^k < 0} \left\{ \frac{\underline{y}_i - y_i^k}{\Delta y_i^k} \right\}, \quad \lambda_k^4 = \min_{i: \Delta y_i^k > 0} \left\{ \frac{\bar{y}_i - y_i^k}{\Delta y_i^k} \right\}.$$

Максимальный шаг вычисляется по формуле

$$\tilde{\lambda}_k = \min\{\lambda_k^1, \lambda_k^2, \lambda_k^3, \lambda_k^4\}.$$

Если $\tilde{\lambda}_k \geq 1$, то точка

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k, \quad \mathbf{y}^{k+1} = \mathbf{y}^k + \Delta\mathbf{y}^k$$

удовлетворяет интервальным ограничениям, а, исходя из (8) и ограничений задачи (9), условию

$$\mathbf{Ax}^{k+1} - \mathbf{y}^{k+1} = \mathbf{0}.$$

Следовательно, вычислительный процесс можно завершить в связи с нахождением допустимого решения системы (5). Если $\tilde{\lambda}_k < 1$, то вычисляем шаг корректировки по правилу

$$\lambda_k = \gamma \tilde{\lambda}_k,$$

где $\gamma \in (0; 2/3)$ - заданная константа.

6. Итеративный переход

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda_k \Delta\mathbf{x}^k, \quad \mathbf{y}^{k+1} = \mathbf{y}^k + \lambda_k \Delta\mathbf{y}^k.$$

Включение в алгоритм процедуры идентификации несовместности

Ключевым новым моментом является включение в схему алгоритма процедуры идентификации несовместности на основе предложенного В.И. Зоркальцевым в [3] критерия. Система (5) несовместна в том и только в том случае, когда существует вектор $\mathbf{u} \in \mathbf{R}^m$, для которого

$$\psi(\mathbf{u}) > 0,$$

где

$$\psi(\mathbf{u}) = \bar{\mathbf{y}}^T \mathbf{u}_- + \underline{\mathbf{y}}^T \mathbf{u}_+ - \bar{\mathbf{x}}^T (\mathbf{A}^T \mathbf{u})_+ - \underline{\mathbf{x}}^T (\mathbf{A}^T \mathbf{u})_-.$$

Соответственно, четвертый пункт алгоритма состоит в следующей проверке: если

$$\psi(\mathbf{u}^k) > 0, \tag{10}$$

где \mathbf{u}^k - вектор множителей Лагранжа ограничений вспомогательной задачи (9), то, согласно приведенному критерию, система (5) несовместна, и, осуществив итеративный переход, расчеты можно завершить, перейдя к новой итерации линеаризации. В то же время несовместность ограничений линеаризованной подзадачи может служить сигналом о возможной несовместности исходной нелинейной системы.

Если система (5) несовместна, то на одной из итераций обязательно будет иметь место неравенство (10). Действительно, двойственной к задаче (7) является следующая:

$$\psi(\mathbf{u}) \rightarrow \max_{\mathbf{u} \in \mathbf{R}^m}, \quad \mathbf{r}^T \mathbf{u} \leq 1. \quad (11)$$

Значения целевых функций для оптимальных решений взаимно-двойственных задач линейного программирования совпадают, поэтому если $\bar{\alpha}$ - оптимальное решение задачи (7), а $\bar{\mathbf{u}}$ - оптимальное решение задачи (11), к которому сходятся векторы \mathbf{u}^k при $k \rightarrow \infty$, то $\psi(\bar{\mathbf{u}}) = \bar{\alpha}$. Следовательно, $\psi(\bar{\mathbf{u}}) > 0$.

Двухвариантная схема решения вспомогательной задачи

Наиболее вычислительно емкой в алгоритме является задача (9) поиска направления корректировки решения. Если решать ее методом Лагранжа, то необходимо сначала вычислить вектор множителей \mathbf{u}^k как решение системы

$$\left(\mathbf{A} \mathbf{X}_k \mathbf{A}^T + \mathbf{Y}_k \right) \mathbf{u} = \mathbf{r}^k, \quad (12)$$

а затем найти направления корректировки переменных

$$\Delta \mathbf{x}^k = \mathbf{X}_k \mathbf{A}^T \mathbf{u}^k, \quad \Delta \mathbf{y}^k = -\mathbf{Y}_k \mathbf{u}^k.$$

При использовании этого пути расчетов основной вычислительной проблемой является решение системы линейных уравнений (12) с симметричной положительно определенной матрицей размерности $m \times m$.

Другой путь состоит в непосредственной подстановке в целевую функцию задачи (9) выражения

$$\Delta \mathbf{y} = \mathbf{A} \Delta \mathbf{x} - \mathbf{r}^k.$$

Получим задачу безусловной минимизации

$$(\Delta \mathbf{x})^T \mathbf{X}_k^{-1} \Delta \mathbf{x} + (\mathbf{A} \Delta \mathbf{x} - \mathbf{r}^k)^T \mathbf{Y}_k^{-1} (\mathbf{A} \Delta \mathbf{x} - \mathbf{r}^k) \rightarrow \min_{\Delta \mathbf{x} \in \mathbf{R}^n}.$$

Приравняв градиент минимизируемой функции нулевому вектору, получим систему линейных уравнений с симметричной положительно определенной матрицей $n \times n$:

$$(\mathbf{X}_k^{-1} + \mathbf{A}^T \mathbf{Y}_k^{-1} \mathbf{A}) \Delta \mathbf{x} = \mathbf{A}^T \mathbf{Y}_k^{-1} \mathbf{r}^k. \quad (13)$$

Найдя из системы (13) вектор $\Delta \mathbf{x}^k$, можем определить $\Delta \mathbf{y}^k$ и \mathbf{u}^k по правилам

$$\Delta \mathbf{y}^k = \mathbf{A} \Delta \mathbf{x}^k - \mathbf{r}^k, \quad \mathbf{u}^k = -\mathbf{Y}_k^{-1} \Delta \mathbf{y}^k.$$

В случае, если $n < m$, вычисления по второму из указанных вариантов могут позволить сократить время расчетов.

Конкретизация вариантов алгоритмов

Два варианта алгоритмов отличаются друг от друга способами вычисления диагональных матриц \mathbf{X}_k и \mathbf{Y}_k весовых коэффициентов. В первом из них, алгоритме F_1 , они рассчитываются по правилам

$$(X_k)_{jj} = \left(\min \{ \bar{x}_j - x_j^k, x_j^k - \underline{x}_j \} \right)^2, \quad j = 1, \dots, n,$$

$$(Y_k)_{ii} = \left(\min \{ \bar{y}_i - y_i^k, y_i^k - \underline{y}_i \} \right)^2, \quad i = 1, \dots, m.$$

Алгоритм F_2 , в отличие от алгоритма F_1 , использует значения множителей Лагранжа, вычисленных на предыдущей итерации. При $k > 1$

$$(X_k)_{jj} = \min \{ \bar{x}_j - x_j^k, x_j^k - \underline{x}_j \} / \max \{ \varepsilon, |v_j^{k-1}| \}, \quad j = 1, \dots, n,$$

$$(Y_k)_{ii} = \min \{ \bar{y}_i - y_i^k, y_i^k - \underline{y}_i \} / \max \{ \varepsilon, |u_i^{k-1}| \}, \quad i = 1, \dots, m.$$

Здесь $\mathbf{v}^{k-1} = \mathbf{A}^T \mathbf{u}^{k-1}$, ε - в простейшем варианте алгоритма заданная малая положительная константа. На первой итерации можно пользоваться правилами алгоритма F_1 .

Варианты F_1 и F_2 являются переформулировками для систем неравенств одноименных алгоритмов, изложенных в предыдущей главе.

Для обоих вариантов прямого алгоритма все компоненты вектора невязок ограничений-равенств монотонно уменьшаются в одной и той же пропорции, исходя из правила

$$\mathbf{r}^{k+1} = (1 - \lambda_k) \mathbf{r}^k, \quad (14)$$

которое является следствием формул алгоритма.

Вместо (8) можно пользоваться этим правилом задания \mathbf{r}^k . Однако с вычислительной точки зрения правило (8) предпочтительнее, поскольку оно одновременно способствует погашению накопленных на предыдущих итерациях погрешностей.

В то же время, выражение (14) наглядно демонстрирует, что максимальный целесообразный шаг λ_k вдоль направления корректировки равен единице.

Двойственная задача

В двойственных алгоритмах осуществляется монотонное по итерациям улучшение решений двойственной задачи линейного программирования внутри области ее допустимых решений. При этом, в результате решения вспомогательной задачи, на каждой итерации вырабатываются векторы, служащие приближением к решению исходной задачи.

Формально можно было бы в качестве двойственной к исходной проблеме использовать задачу (11). Однако есть резон в исключении ограничения-неравенства из формулировки двойственной задачи. Вектор \mathbf{r} в этом ограничении зависит от выбора исходного приближения $(\mathbf{x}^1, \mathbf{y}^1)$.

Желательно иметь формулировку двойственной задачи, инвариантную к этому выбору, не связанную с ним.

Отметим, что ограничение не является активным для оптимального решения задачи (11), если исходная система неравенств (5) совместная. То есть для этого случая оно практически лишнее. Оно является существенным в случае, если исходная система (5) несовместна. Но этот случай идентифицируется благодаря введенному критерию несовместности. Поэтому данное ограничение можно безболезненно исключить. На основе преобразований

$$\mathbf{w}^1 = -\mathbf{u}_-, \quad \mathbf{w}^2 = \mathbf{u}_+, \quad \mathbf{v}^1 = \left(\mathbf{A}^T \mathbf{u}\right)_+, \quad \mathbf{v}^2 = -\left(\mathbf{A}^T \mathbf{u}\right)_-$$

перейдем к задаче линейного программирования относительно переменных $\mathbf{v}^1, \mathbf{v}^2 \in \mathbf{R}^n$, $\mathbf{w}^1, \mathbf{w}^2 \in \mathbf{R}^m$

$$\bar{\mathbf{y}}^T \mathbf{w}^1 - \underline{\mathbf{y}}^T \mathbf{w}^2 + \bar{\mathbf{x}}^T \mathbf{v}^1 - \underline{\mathbf{x}}^T \mathbf{v}^2 \rightarrow \min, \quad (15)$$

$$\mathbf{A}^T (\mathbf{w}^1 - \mathbf{w}^2) + \mathbf{v}^1 - \mathbf{v}^2 = \mathbf{0}, \quad (16)$$

$$\mathbf{w}^1 \geq \mathbf{0}, \quad \mathbf{w}^2 \geq \mathbf{0}, \quad \mathbf{v}^1 \geq \mathbf{0}, \quad \mathbf{v}^2 \geq \mathbf{0}. \quad (17)$$

Из выражения целевой функции несложно соотнести переменные введенной задачи (15)-(17) с ограничениями исходной задачи. Векторы \mathbf{w}^1 и \mathbf{w}^2 могут интерпретироваться как двойственные оценки ограничений $\bar{\mathbf{y}} \geq \mathbf{u}$ и $\mathbf{u} \geq \underline{\mathbf{y}}$, а векторы \mathbf{v}^1 и \mathbf{v}^2 - соответственно как двойственные оценки ограничений $\bar{\mathbf{x}} \geq \mathbf{x}$ и $\mathbf{x} \geq \underline{\mathbf{x}}$.

Поиск стартового приближения

Исходное приближение должно удовлетворять ограничениям (16) и (17). Причем неравенства (17) должны выполняться в строгой форме. Таким приближением может служить любая точка $\mathbf{v}^{1,1} = \mathbf{v}^{2,1} > \mathbf{0}$, $\mathbf{w}^{1,1} = \mathbf{w}^{2,1} > \mathbf{0}$. В частности, в первоначальной редакции алгоритма было задано

$$v_j^{1,1} = v_j^{2,1} = 1, \quad j = 1, \dots, n, \quad w_i^{1,1} = w_i^{2,1} = 1, \quad i = 1, \dots, m.$$

Такое начальное приближение универсально, однако оно ни в коей мере не учитывает специфику задачи. Поскольку переменные \mathbf{v}^1 и \mathbf{v}^2 , являясь двойственными оценками, в некотором смысле, обратно пропорциональны разностям $\bar{\mathbf{x}} - \mathbf{x}$ и $\mathbf{x} - \underline{\mathbf{x}}$ соответственно (аналогичная ситуация с \mathbf{w}^1 и \mathbf{w}^2), целесообразнее проводить инициализацию алгоритма, используя следующие правила:

$$v_j^{1,1} = v_j^{2,1} = 2/(\bar{x}_j - \underline{x}_j), \quad j = 1, \dots, n, \quad w_i^{1,1} = w_i^{2,1} = 2/(\bar{y}_i - \underline{y}_i), \quad i = 1, \dots, m.$$

Двойственные алгоритмы

Перейдем непосредственно к описанию алгоритмов. Вычислительный процесс на каждой итерации $k = 1, 2, \dots$ состоит из следующих операций:

1. Вычисление диагональных матриц \mathbf{V}_k^1 , \mathbf{V}_k^2 , \mathbf{W}_k^1 и \mathbf{W}_k^2 весовых коэффициентов, конкретизируемых ниже.

2. Нахождение векторов \mathbf{x}^k и \mathbf{y}^k как решения следующей вспомогательной задачи

$$\begin{aligned} & (\bar{\mathbf{x}} - \mathbf{x})^T \mathbf{V}_k^1 (\bar{\mathbf{x}} - \mathbf{x}) + (\mathbf{x} - \underline{\mathbf{x}})^T \mathbf{V}_k^2 (\mathbf{x} - \underline{\mathbf{x}}) + \\ & + (\bar{\mathbf{y}} - \mathbf{y})^T \mathbf{W}_k^1 (\bar{\mathbf{y}} - \mathbf{y}) + (\mathbf{y} - \underline{\mathbf{y}})^T \mathbf{W}_k^2 (\mathbf{y} - \underline{\mathbf{y}}) \rightarrow \min_{\mathbf{x} \in \mathbf{R}^n, \mathbf{y} \in \mathbf{R}^m} \end{aligned} \quad (18)$$

при ограничениях

$$\mathbf{A}\mathbf{x} - \mathbf{y} = \mathbf{0}. \quad (19)$$

Если выполняются неравенства $\bar{\mathbf{x}} \geq \mathbf{x}^k \geq \underline{\mathbf{x}}$ и $\bar{\mathbf{y}} \geq \mathbf{y}^k \geq \underline{\mathbf{y}}$, то расчеты можно завершить в связи с получением допустимого решения системы (5).

В этом же пункте попутно вычисляется вектор \mathbf{u}^k множителей Лагранжа ограничений (19).

3. Первая проверка несовместности системы (5). Если выполняется условие $\psi(\mathbf{u}^k) < 0$, то вычисления можно завершить и перейти к новой итерации линеаризации в связи с идентификацией несовместности системы (5).

4. Нахождение направлений корректировки переменных двойственной задачи:

$$\begin{aligned}\Delta \mathbf{v}^{1,k} &= -\mathbf{V}_k^1(\bar{\mathbf{x}} - \mathbf{x}^k), & \Delta \mathbf{v}^{2,k} &= -\mathbf{V}_k^2(\mathbf{x}^k - \underline{\mathbf{x}}), \\ \Delta \mathbf{w}^{1,k} &= -\mathbf{W}_k^1(\bar{\mathbf{y}} - \mathbf{y}^k), & \Delta \mathbf{w}^{2,k} &= -\mathbf{W}_k^2(\mathbf{y}^k - \underline{\mathbf{y}}).\end{aligned}$$

5. Вычисление шага корректировки переменных двойственной задачи.

Для этого ищутся вспомогательные величины

$$\begin{aligned}\lambda_k^1 &= \min_{j: \Delta v_j^{1,k} < 0} \left\{ \frac{v_j^{1,k}}{-\Delta v_j^{1,k}} \right\}, & \lambda_k^2 &= \min_{j: \Delta v_j^{2,k} < 0} \left\{ \frac{v_j^{2,k}}{-\Delta v_j^{2,k}} \right\}, \\ \lambda_k^3 &= \min_{i: \Delta w_i^{1,k} < 0} \left\{ \frac{w_i^{1,k}}{-\Delta w_i^{1,k}} \right\}, & \lambda_k^4 &= \min_{i: \Delta w_i^{2,k} < 0} \left\{ \frac{w_i^{2,k}}{-\Delta w_i^{2,k}} \right\}.\end{aligned}$$

Шаг корректировки вычисляется по формуле

$$\lambda_k = \gamma \cdot \min\{\lambda_k^1, \lambda_k^2, \lambda_k^3, \lambda_k^4\}, \quad \gamma \in (0; 2/3).$$

6. Итеративный переход, осуществляемый по формулам

$$\begin{aligned}\mathbf{w}^{1,k+1} &= \mathbf{w}^{1,k} + \lambda_k \Delta \mathbf{w}^{1,k}, & \mathbf{w}^{2,k+1} &= \mathbf{w}^{2,k} + \lambda_k \Delta \mathbf{w}^{2,k}, \\ \mathbf{v}^{1,k+1} &= \mathbf{v}^{1,k} + \lambda_k \Delta \mathbf{v}^{1,k}, & \mathbf{v}^{2,k+1} &= \mathbf{v}^{2,k} + \lambda_k \Delta \mathbf{v}^{2,k}.\end{aligned}$$

7. Вторая проверка несовместности. Если выполняется неравенство $\psi(\mathbf{w}^{2,k} - \mathbf{w}^{1,k}) < 0$, то система (5) несовместна, в связи с чем можем перейти к новой итерации линеаризации.

Двухвариантная схема решения вспомогательной задачи

При решении задачи (18)-(19) снова можем действовать двойко в зависимости от соотношения размерностей m и n . Если $m < n$, задачу будем решать методом Лагранжа. Для начала рассчитаем вспомогательные диагональные матрицы

$$\mathbf{V}_k^3 = (\mathbf{V}_k^1 + \mathbf{V}_k^2)^{-1}, \quad \mathbf{W}_k^3 = (\mathbf{W}_k^1 + \mathbf{W}_k^2)^{-1}.$$

Затем найдем вектор множителей Лагранжа \mathbf{u}^k ограничений (19):

$$\mathbf{u}^k = \left(\mathbf{A} \mathbf{V}_3^k \mathbf{A}^T + \mathbf{W}_3^k \right)^{-1} \left(\mathbf{W}_k^3 \left(\mathbf{W}_k^1 \bar{\mathbf{y}} + \mathbf{W}_k^2 \underline{\mathbf{y}} \right) - \mathbf{A} \mathbf{V}_3^k \left(\mathbf{V}_k^1 \bar{\mathbf{x}} + \mathbf{V}_k^2 \underline{\mathbf{x}} \right) \right).$$

Здесь наиболее вычислительно емкой процедурой будет обращение положительно определенной симметричной матрицы размерности $m \times m$. И в заключение вычислим искомые векторы

$$\mathbf{x}^k = \mathbf{V}_k^3 \left(\mathbf{V}_k^1 \bar{\mathbf{x}} + \mathbf{V}_k^2 \underline{\mathbf{x}} + \mathbf{A}^T \mathbf{u}^k \right), \quad \mathbf{y}^k = \mathbf{W}_k^3 \left(\mathbf{W}_k^1 \bar{\mathbf{y}} + \mathbf{W}_k^2 \underline{\mathbf{y}} - \mathbf{u}^k \right) = \mathbf{A} \mathbf{x}^k.$$

Второй вариант (его полезно использовать при $m > n$) основывается на подстановке равенства $\mathbf{y} = \mathbf{A} \mathbf{x}$ в минимизируемую функцию (18), что приводит задачу к проблеме безусловной минимизации квадратичной функции только от вектора \mathbf{x} . Приравняв градиент нулевому вектору, получим систему линейных уравнений с матрицей размерности $n \times n$

$$\left(\mathbf{V}_k^1 + \mathbf{V}_k^2 + \mathbf{A}^T \mathbf{W}_k^1 \mathbf{A} + \mathbf{A}^T \mathbf{W}_k^2 \mathbf{A} \right) \mathbf{x} = \mathbf{V}_k^1 \bar{\mathbf{x}} + \mathbf{V}_k^2 \underline{\mathbf{x}} + \mathbf{A}^T \mathbf{W}_k^1 \bar{\mathbf{y}} + \mathbf{A}^T \mathbf{W}_k^2 \underline{\mathbf{y}}.$$

Найдя из нее \mathbf{x}^k , вычисляем $\mathbf{y}^k = \mathbf{A} \mathbf{x}^k$ и вектор множителей Лагранжа ограничений (19) по формуле

$$\mathbf{u}^k = \mathbf{W}_k^1 (\bar{\mathbf{y}} - \mathbf{y}) - \mathbf{W}_k^2 (\mathbf{y} - \underline{\mathbf{y}}).$$

Конкретизация вариантов алгоритмов

Отличие алгоритмов выражается в различных правилах вычисления диагональных матриц весовых коэффициентов. В алгоритме G_1 они вычисляются в соответствии со следующими правилами:

$$\begin{aligned} \left(V_k^1 \right)_{jj} &= \left(v_j^{1,k} \right)^2, & \left(V_k^2 \right)_{jj} &= \left(v_j^{2,k} \right)^2, & j &= 1, \dots, n, \\ \left(W_k^1 \right)_{ii} &= \left(w_i^{1,k} \right)^2, & \left(W_k^2 \right)_{ii} &= \left(w_i^{2,k} \right)^2, & i &= 1, \dots, m. \end{aligned}$$

В алгоритме G_2 , начиная со второй итерации, они вычисляются по формулам

$$\begin{aligned} \left(V_k^1 \right)_{jj} &= v_j^{1,k} / \max \left\{ \varepsilon, \bar{x}_j - x_j^k \right\}, & j &= 1, \dots, n, \\ \left(V_k^2 \right)_{jj} &= v_j^{2,k} / \max \left\{ \varepsilon, x_j^k - \underline{x}_j \right\}, & j &= 1, \dots, n, \end{aligned}$$

$$\begin{aligned} (W_k^1)_{ii} &= w_i^{1,k} / \max\{\varepsilon, \bar{y}_i - y_i^k\}, \quad i = 1, \dots, m, \\ (W_k^2)_{ii} &= w_i^{2,k} / \max\{\varepsilon, y_i^k - \underline{y}_i\}, \quad i = 1, \dots, m. \end{aligned}$$

при некотором малом $\varepsilon > 0$. При $k = 1$ можно положить $(\mathbf{x}^1, \mathbf{y}^1)$ равным исходному приближению в прямых алгоритмах, то есть использовать векторы, удовлетворяющие условию (6). В качестве другого варианта можно предложить пользоваться на первой итерации правилами алгоритма G_1 .

Проблема выбора значения ε для алгоритма G_2 оказывается очень важной. В его первоначальной версии ε использовалось лишь для защиты от деления на ноль и принималось равным 10^{-10} . Однако на отдельных задачах такая схема показала себя не лучшим образом. Рассмотрим, из-за чего это может происходить.

В алгоритме G_2 всегда, пока не найдено решение, часть переменных выходит за пределы допустимого интервала. Всем им в соответствии с данной схемой должны соответствовать огромные величины весовых коэффициентов. Подобные “штрафы” должны, по идее алгоритма, стимулировать возвращение переменных, вышедших из допустимой области, внутрь интервала. Однако, жестко “вгоняя” одни переменные внутрь интервала, мы получаем ситуацию, когда другие переменные (которым соответствуют очень небольшие, в сравнении, весовые коэффициенты) начинают выходить вовне.

Можно задавать значение ε , руководствуясь спецификой задачи. Это значение может изменяться от итерации к итерации. Главный принцип должен состоять в том, чтобы переменным, вышедшим за границы допустимого интервала соответствовали значения весовых коэффициентов, превосходящие значения весовых коэффициентов, соответствующих переменным, находящимся внутри. В частности, в модифицированном

варианте алгоритма вычисление значения ε ведется по следующему предлагаемому правилу:

$$\varepsilon = k \min \left\{ \begin{array}{l} \min_{j: x_j^{k-1} < \bar{x}_j - \underline{\varepsilon}} (\bar{x}_j - x_j^{k-1}), \quad \min_{j: x_j^{k-1} > \underline{x}_j + \underline{\varepsilon}} (x_j^{k-1} - \underline{x}_j), \\ \min_{i: y_i^{k-1} < \bar{y}_i - \underline{\varepsilon}} (\bar{y}_i - y_i^{k-1}), \quad \min_{i: y_i^{k-1} > \underline{y}_i + \underline{\varepsilon}} (y_i^{k-1} - \underline{y}_i) \end{array} \right\},$$

где $\underline{\varepsilon} = 10^{-10}$, $k = 0.2$.

Прямо-двойственный алгоритм Н

В прямых алгоритмах F_1 и F_2 приближения $(\mathbf{x}^k, \mathbf{y}^k)$ на каждой итерации $k = 1, 2, \dots$ удовлетворяют интервальным ограничениям-неравенствам, но не удовлетворяют балансовым ограничениям-равенствам. В двойственных алгоритмах G_1 и G_2 - напротив, удовлетворяют ограничениям-равенствам, но могут выходить за границы допустимых интервалов. В рассматриваемом в данном параграфе прямо-двойственном алгоритме Н сочетаются обе эти траектории.

Обозначим $(\tilde{\mathbf{x}}^1, \tilde{\mathbf{y}}^1)$ начальное приближение, которое удовлетворяют неравенствам (6). Далее действуем в соответствии с правилами алгоритма G_2 с тем отличием, что в формулах вычисления весовых коэффициентов вместо векторов $\mathbf{x}^{k-1}, \mathbf{y}^{k-1}$ используем векторы $\tilde{\mathbf{x}}^k, \tilde{\mathbf{y}}^k$, которые итеративно пересчитываются по формулам

$$\tilde{\mathbf{x}}^{k+1} = \tilde{\mathbf{x}}^k + \gamma \lambda_k \Delta \mathbf{x}^k, \quad \tilde{\mathbf{y}}^{k+1} = \tilde{\mathbf{y}}^k + \gamma \lambda_k \Delta \mathbf{y}^k,$$

где направления корректировки определяются формулами

$$\Delta \mathbf{x}^k = \mathbf{x}^k - \tilde{\mathbf{x}}^k, \quad \Delta \mathbf{y}^k = \mathbf{y}^k - \tilde{\mathbf{y}}^k,$$

а шаг корректировки λ_k находится, исходя из правил

$$\lambda_k^1 = \min_{j: \Delta x_j^k < 0} \left\{ \frac{x_j - \tilde{x}_j^k}{\Delta x_j^k} \right\}, \quad \lambda_k^2 = \min_{j: \Delta x_j^k > 0} \left\{ \frac{\bar{x}_j - \tilde{x}_j^k}{\Delta x_j^k} \right\},$$

$$\lambda_k^3 = \min_{i: \Delta y_i^k < 0} \left\{ \frac{y_i - \tilde{y}_i^k}{\Delta y_i^k} \right\}, \quad \lambda_k^4 = \min_{i: \Delta y_i^k > 0} \left\{ \frac{\bar{y}_i - \tilde{y}_i^k}{\Delta y_i^k} \right\},$$

$$\lambda_k = \min \{1, \lambda_k^1, \lambda_k^2, \lambda_k^3, \lambda_k^4\}.$$

Если $\lambda_k = 1$, то точка $\mathbf{x}^{k+1} = \tilde{\mathbf{x}}^k + \Delta \mathbf{x}^k$, $\mathbf{y}^{k+1} = \tilde{\mathbf{y}}^k + \Delta \mathbf{y}^k$ является допустимой для системы (5), в связи с чем вычислительный процесс и завершается.

Прямо-двойственный алгоритм обеспечивает уменьшение невязок по итерациям в соответствии с формулами

$$\tilde{\mathbf{r}}^{k+1} = (1 - \gamma \lambda_k) \tilde{\mathbf{r}}^k,$$

где

$$\tilde{\mathbf{r}}^k = \tilde{\mathbf{y}}^k - \mathbf{A} \tilde{\mathbf{x}}^k.$$

Отметим, что экспериментально данный алгоритм будет исследоваться впервые.

Экспериментальное исследование

Целью проведенного экспериментального исследования является проверка работоспособности рассмотренных алгоритмов, сопоставление их по скорости сходимости, а также проверка способности быстро выявлять несовместные системы на основе приведенного критерия.

Все предложенные алгоритмы протестируем на 14 совместных и 14 несовместных (так для краткости будем называть задачи с несовместными ограничениями) задачах размерности от 30×80 до 41×80 , полученных из моделей управления режимами электроэнергетических систем. Исследуемые электроэнергетические системы содержат гораздо большее количество узлов (как правило, несколько сотен), однако благодаря специальной процедуре на каждой итерации учитывается только небольшая часть ограничений.

Также протестируем алгоритмы на следующем тестовом примере (n - нечетное, брались значения $n = 19$ и $n = 201$):

$$x_i \in [0, n], \quad i = 1, \dots, n,$$

$$y_i = x_{i+1} - x_i \in [-1, 1], \quad i = 1, \dots, n-1,$$

$$y_n = -x_1 + x_{(n+1)/2} - x_n \in [(n-1)/2 - \varepsilon, n], \quad \varepsilon = 0.0001.$$

Допустимыми в нем будут, например, векторы x , заданные по правилам

$$x_i = i - 1, \quad i = 1, \dots, (n+1)/2,$$

$$x_i = n - i, \quad i = (n+3)/2, \dots, n.$$

Поскольку все алгоритмы имеют примерно одинаковый объем вычислений на итерации, характеристикой объема вычислений можно считать число итераций, за которое алгоритм находит допустимое решение системы (5), либо выявляет, что она несовместна. Чтобы не загромождать таблицу большим количеством цифр, занесем в нее среднее количество итераций, необходимое каждому алгоритму для решения 14 совместных и 14 несовместных задач. В скобках запишем минимальное и максимальное число итераций, необходимых для решения отдельной задачи.

Таблица 4. Среднее число итераций, необходимое для решения задач

Алгоритм	Совмест.задачи (30 × 80) – (41 × 80)	Несовмест.задачи (30 × 80) – (41 × 80)	Тест.пример (19 × 19)	Тест.пример (201 × 201)
F₁	9.5 (3-13)	1.6 (1-5)	13	19
F₂	13.4 (7-17)	1.6 (1-7)	8	7
G₁	10.2 (6-13)	1 (все 1)	11	16
G₂	5.9 (3-8)	1 (все 1)	6	9
H	9.6 (4-13)	1 (все 1)	13	17

Таблица наглядно демонстрирует практическую эффективность критерия несовместности. Несовместность обычно выявляется на первой же итерации. Особенно хорошо здесь зарекомендовали себя двойственные и прямо-двойственный алгоритмы. Также алгоритмы продемонстрировали высокие скоростные характеристики при решении совместных задач.

В то же время, как показали экспериментальные расчеты, двойственные алгоритмы, демонстрировавшие здесь наилучшие результаты, при решении исходной нелинейной задачи поиска допустимых режимов ЭЭС оказались в конечном итоге менее эффективными. Основной причиной этого является наличие хорошего исходного приближения для прямых алгоритмов - полученного на предыдущей глобальной итерации вектора \mathbf{x}^t , в то время как двойственные алгоритмы никак не учитывают информацию с предыдущей глобальной итерации.

Таким образом, несмотря на более быстрое решение линеаризованных задач двойственным алгоритмам нередко требовалось больше итераций линеаризации, а иногда и больше времени для решения исходной проблемы.

3.3. Алгоритмы центрального и скошенного пути применительно к решению линеаризованной подзадачи

В данном разделе будем адаптировать к решению системы линейных уравнений и неравенств (5) алгоритмы центрального пути. Лучшие из них, как было показано в главе 2, продемонстрировали хорошие результаты на паре взаимно-двойственных задач линейного программирования. Интервальные ограничения на переменные в системе (5) позволяют успешно решить проблему инициализации алгоритмов. Кроме того, критерий несовместности также успешно вписывается в алгоритмическую схему. Эти особенности дают алгоритмам центрального пути дополнительные преимущества именно на системах линейных уравнений и неравенств.

Центрирование и нормировка переменных

Проведем центрирование и нормировку переменных по правилам

$$x'_j = 2 \frac{\bar{x}_j - \underline{x}_j}{\bar{x}_j - \underline{x}_j}, \quad j = 1, \dots, n, \quad y'_i = 2 \frac{\bar{y}_i - \underline{y}_i}{\bar{y}_i - \underline{y}_i}, \quad i = 1, \dots, m.$$

При этом задача (5) сведется к следующей:

$$\mathbf{A}'\mathbf{x}' - \mathbf{y}' = \mathbf{b}, \quad 2\mathbf{e} \geq \mathbf{x}' \geq \mathbf{0}, \quad 2\mathbf{e} \geq \mathbf{y}' \geq \mathbf{0}. \quad (20)$$

Здесь \mathbf{e} - вектор из единиц соответствующей размерности, элементы матрицы \mathbf{A}' и вектора \mathbf{b} вычисляются по следующим формулам:

$$a'_{ij} = a_{ij} \frac{\bar{x}_j - \underline{x}_j}{\bar{y}_i - \underline{y}_i}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \quad b_i = 2 \frac{\underline{y}_i - \sum_{j=1}^n a_{ij} \underline{x}_j}{\bar{y}_i - \underline{y}_i}, \quad i = 1, \dots, m.$$

Получив допустимое решение $(\mathbf{x}', \mathbf{y}')$ задачи (20), можно с помощью линейных преобразований перейти к исходным переменным (\mathbf{x}, \mathbf{y}) :

$$x_j = \underline{x}_j + x'_j \frac{\bar{x}_j - \underline{x}_j}{2}, \quad j = 1, \dots, n, \quad y_i = \underline{y}_i + y'_i \frac{\bar{y}_i - \underline{y}_i}{2}, \quad i = 1, \dots, m.$$

В дальнейшем будем оперировать системой уравнений и неравенств, записанной в форме (20). Для сокращения записи опустим штрихи над матрицей \mathbf{A} и переменными.

Переход к задаче линейного программирования

Чтобы избавиться от неравенств $\mathbf{x} \leq 2\mathbf{e}$ и $\mathbf{y} \leq 2\mathbf{e}$, введем неотрицательные переменные $\mathbf{s} \in \mathbf{R}^n$ и $\mathbf{q} \in \mathbf{R}^m$, такие что по всем компонентам выполняются равенства $x_j + s_j = 2$, $y_i + q_i = 2$. В частности, имеем начальное приближение $\mathbf{x}^1 = \mathbf{s}^1 = \mathbf{e}$, $\mathbf{y}^1 = \mathbf{q}^1 = \mathbf{e}$. Обозначим за вектор \mathbf{r} невязку ограничений-равенств $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}^1 + \mathbf{y}^1$.

С помощью уже рассмотренного стандартного приема введения дополнительной переменной перейдем от системы (20) к задаче линейного программирования. Справа от черты выпишем двойственные оценки ограничений.

объема требуемых вычислений), справедливые для стандартной пары задач, переносятся сюда. Тем не менее, расписав алгоритмы непосредственно для пары задач (21)-(22), можно существенно уменьшить объем требуемых вычислений.

Центральный путь

Рассмотрим задачу минимизации прямой логарифмической барьерной функции на допустимом множестве

$$\alpha - \mu \sum_{j=1}^n \ln x_j - \mu \sum_{j=1}^n \ln s_j - \mu \sum_{i=1}^m \ln y_i - \mu \sum_{i=1}^m \ln q_i - \mu \ln \alpha \rightarrow \min, \quad (23)$$

$$\mathbf{Ax} - \mathbf{y} + \alpha \mathbf{r} = \mathbf{b}, \quad -\mathbf{x} - \mathbf{s} = -2\mathbf{e}, \quad -\mathbf{y} - \mathbf{q} = -2\mathbf{e}. \quad (24)$$

Для любого положительного μ существует единственная точка, принадлежащая допустимому множеству пары задач (21)-(22), для которой по всем компонентам выполняются условия

$$x_j(\mathbf{v} - \mathbf{A}^T \mathbf{u})_j = \mu, \quad s_j v_j = \mu, \quad y_i(\mathbf{u} + \mathbf{w})_i = \mu, \quad q_i w_i = \mu, \quad \alpha(1 - \mathbf{r}^T \mathbf{u}) = \mu. \quad (25)$$

Переменные прямой задачи \mathbf{x} , \mathbf{s} , \mathbf{y} , \mathbf{q} и α являются оптимальным решением задачи (23)-(24), переменные двойственной задачи \mathbf{u} , \mathbf{v} и \mathbf{w} - множителями Лагранжа ограничений (24).

Центральный путь образует множество точек $\mathbf{x}(\mu)$, $\mathbf{s}(\mu)$, $\mathbf{y}(\mu)$, $\mathbf{q}(\mu)$, $\alpha(\mu)$, $\mathbf{u}(\mu)$, $\mathbf{v}(\mu)$ и $\mathbf{w}(\mu)$ при всех $\mu > 0$, для которых выполняются условия (25). При $\mu \rightarrow 0$ точки центрального пути сходятся к оптимальным решениям задач (21)-(22).

Благодаря особой структуре задачи, имеется начальное приближение, являющееся точкой центрального пути:

$$\begin{aligned} x_j^1 = s_j^1 = 1, \quad j = 1, \dots, n, & \quad y_i^1 = q_i^1 = 1, \quad i = 1, \dots, m, & \quad \alpha^1 = 1, \\ u_i^1 = 0, \quad i = 1, \dots, m, & \quad v_j^1 = 1, \quad j = 1, \dots, n, & \quad w_i^1 = 1, \quad i = 1, \dots, m. \end{aligned}$$

Очевидно, что вместе с $\mu^1 = 1$ это приближение удовлетворяет условию (25).

Конус центрального пути

Конус центрального пути для пары задач (21)-(22) будет состоять из всех относительно внутренних точек множеств их допустимых решений, для которых выполняется условие

$$\begin{aligned} & \sum_{j=1}^n \left(\mu - x_j (\mathbf{v} - \mathbf{A}^T \mathbf{u})_j \right)^2 + \sum_{j=1}^n (\mu - s_j v_j)^2 + \\ & + \sum_{i=1}^m (\mu - y_i (\mathbf{u} + \mathbf{w})_i)^2 + \sum_{i=1}^m (\mu - q_i w_i)^2 + \left(\mu - \alpha (1 - \mathbf{r}^T \mathbf{u}) \right)^2 \leq \theta \mu^2. \end{aligned} \quad (26)$$

Здесь, как и прежде, $\sqrt{\theta} \in (0;1)$ - радиус конуса центрального пути.

Суть алгоритмов оптимизации в конусе центрального пути заключается в построении последовательностей точек $(\mathbf{x}^k, \mathbf{s}^k, \mathbf{y}^k, \mathbf{q}^k, \alpha^k, \mathbf{u}^k, \mathbf{v}^k, \mathbf{w}^k)$ и соответствующих значений μ^k , $k = 1, 2, \dots$, таких что для них выполняется условие (26), и при этом значение параметра центрального пути μ монотонно уменьшается в соответствии с неравенством

$$\mu^{k+1} \leq \left(1 - 0.5 / \sqrt{2n + 2m + 1} \right) \mu^k. \quad (27)$$

Метод Ньютона

Перейдем непосредственно к алгоритмам. Применим для решения задачи (23)-(24) метод Ньютона в точке $(\mathbf{x}^k, \mathbf{s}^k, \mathbf{y}^k, \mathbf{q}^k, \alpha^k, \mu^k)$. Квадратичная аппроксимация минимизируемой логарифмической барьерной функции будет иметь вид

$$\begin{aligned} & - \mu^k (\mathbf{X}^k)^{-1} \Delta \mathbf{x} + \frac{\mu^k}{2} (\mathbf{X}^k)^{-2} (\Delta \mathbf{x})^2 - \mu^k (\mathbf{S}^k)^{-1} \Delta \mathbf{s} + \frac{\mu^k}{2} (\mathbf{S}^k)^{-2} (\Delta \mathbf{s})^2 - \\ & - \mu^k (\mathbf{Y}^k)^{-1} \Delta \mathbf{y} + \frac{\mu^k}{2} (\mathbf{Y}^k)^{-2} (\Delta \mathbf{y})^2 - \mu^k (\mathbf{Q}^k)^{-1} \Delta \mathbf{q} + \frac{\mu^k}{2} (\mathbf{Q}^k)^{-2} (\Delta \mathbf{q})^2 + \\ & + \Delta \alpha - \frac{\mu^k}{\alpha^k} \Delta \alpha + \frac{\mu^k}{2(\alpha^k)^2} (\Delta \alpha)^2 \rightarrow \min_{\Delta \mathbf{x}, \Delta \mathbf{s}, \Delta \mathbf{y}, \Delta \mathbf{q}, \Delta \alpha}. \end{aligned}$$

Ограничения-равенства выглядят следующим образом:

$$\begin{array}{l|l}
\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{y} + \Delta\alpha\mathbf{r} = \mathbf{0}, & \mathbf{u} \\
-\Delta\mathbf{x} - \Delta\mathbf{s} = \mathbf{0}, & \mathbf{v} \\
-\Delta\mathbf{y} - \Delta\mathbf{q} = \mathbf{0}. & \mathbf{w}
\end{array} \quad (28)$$

Дадим некоторые пояснения. Векторами $\Delta\mathbf{x}, \Delta\mathbf{s}, \Delta\mathbf{y}, \Delta\mathbf{q}, \Delta\alpha$ обозначены направления корректировки. $\mathbf{X}^k, \mathbf{S}^k, \mathbf{Y}^k, \mathbf{Q}^k$ - соответствующие векторам $\mathbf{x}, \mathbf{s}, \mathbf{y}, \mathbf{q}$, диагональные матрицы. За чертой в (28) выписаны множители Лагранжа ограничений.

Применяя метод Лагранжа, получим следующую систему уравнений:

$$\begin{aligned}
-\mu^k (\mathbf{X}^k)^{-1} + \mu^k (\mathbf{X}^k)^{-2} \Delta\mathbf{x} &= \mathbf{A}^T \mathbf{u} - \mathbf{v}, \\
-\mu^k (\mathbf{S}^k)^{-1} + \mu^k (\mathbf{S}^k)^{-2} \Delta\mathbf{s} &= -\mathbf{v}, \\
-\mu^k (\mathbf{Y}^k)^{-1} + \mu^k (\mathbf{Y}^k)^{-2} \Delta\mathbf{y} &= -\mathbf{u} - \mathbf{w}, \\
-\mu^k (\mathbf{Q}^k)^{-1} + \mu^k (\mathbf{Q}^k)^{-2} \Delta\mathbf{q} &= -\mathbf{w}, \\
-\frac{\mu^k}{\alpha^k} + \frac{\mu^k}{(\alpha^k)^2} \Delta\alpha &= \mathbf{r}^T \mathbf{u} - 1.
\end{aligned} \quad (29)$$

Решая совместно (28)-(29), получим формулы для пересчета переменных двойственной задачи:

$$\begin{aligned}
\mathbf{u}^{k+1} &= \left(\mathbf{A}\mathbf{D}_1^k \mathbf{A}^T + \mathbf{D}_2^k + (\alpha^k)^2 \mathbf{r}\mathbf{r}^T \right)^{-1} \times \\
&\quad \times \left((\alpha^k)^2 \mathbf{r} - \mu^k \mathbf{b} + 2\mu^k \mathbf{A}(\mathbf{X}^k)^2 \mathbf{D}_1^k \mathbf{e} - 2\mu^k (\mathbf{Y}^k)^2 \mathbf{D}_2^k \mathbf{e} \right), \\
v_j^{k+1} &= \frac{2\mu^k + (x_j^k)^2 (\mathbf{A}^T \mathbf{u}^{k+1})_j}{(x_j^k)^2 + (s_j^k)^2}, \quad w_i^{k+1} = \frac{2\mu^k - (y_i^k)^2 u_i^{k+1}}{(y_i^k)^2 + (q_i^k)^2},
\end{aligned} \quad (30)$$

$$\mathbf{D}_1^k = \left((\mathbf{X}^k)^{-2} + (\mathbf{S}^k)^{-2} \right)^{-1}, \quad \mathbf{D}_2^k = \left((\mathbf{Y}^k)^{-2} + (\mathbf{Q}^k)^{-2} \right)^{-1}.$$

Направления корректировки переменных прямой задачи вычисляем по правилам

$$\begin{aligned}
\Delta x_j^k &= x_j^k - \frac{1}{\mu^k} (x_j^k)^2 (\mathbf{v}^{\mathbf{k}+1} - \mathbf{A}^T \mathbf{u}^{\mathbf{k}+1})_j, \quad j = 1, \dots, n, \\
\Delta s_j^k &= s_j^k - \frac{1}{\mu^k} (s_j^k)^2 v_j^{\mathbf{k}+1}, \quad j = 1, \dots, n, \\
\Delta y_i^k &= y_i^k - \frac{1}{\mu^k} (y_i^k)^2 (\mathbf{u}^{\mathbf{k}+1} + \mathbf{w}^{\mathbf{k}+1})_i, \quad i = 1, \dots, m, \\
\Delta q_i^k &= q_i^k - \frac{1}{\mu^k} (q_i^k)^2 w_i^{\mathbf{k}+1}, \quad i = 1, \dots, m, \\
\Delta \alpha^k &= \alpha^k - \frac{1}{\mu^k} (\alpha^k)^2 (1 - \mathbf{r}^T \mathbf{u}).
\end{aligned} \tag{31}$$

Пересчет осуществляется следующим образом:

$$\begin{aligned}
\mathbf{x}^{\mathbf{k}+1} &= \mathbf{x}^{\mathbf{k}} + \Delta \mathbf{x}^{\mathbf{k}}, \quad \mathbf{s}^{\mathbf{k}+1} = \mathbf{s}^{\mathbf{k}} + \Delta \mathbf{s}^{\mathbf{k}}, \\
\mathbf{y}^{\mathbf{k}+1} &= \mathbf{y}^{\mathbf{k}} + \Delta \mathbf{y}^{\mathbf{k}}, \quad \mathbf{q}^{\mathbf{k}+1} = \mathbf{q}^{\mathbf{k}} + \Delta \mathbf{q}^{\mathbf{k}}, \\
\alpha^{\mathbf{k}+1} &= \alpha^{\mathbf{k}} + \Delta \alpha^{\mathbf{k}}.
\end{aligned} \tag{32}$$

Простейший вариант алгоритма - алгоритм А (обозначения те же, что и для алгоритмов решения задач линейного программирования, введенных во второй главе) заключается в том, чтобы далее пересчитать значение параметра центрального пути μ по правилу

$$\mu^{\mathbf{k}+1} = \left(1 - 0.5/\sqrt{2n + 2m + 1}\right) \mu^{\mathbf{k}}.$$

Далее будут рассматриваться усовершенствованные варианты алгоритмов, а именно модификации алгоритмов С, D и E, адаптированные к паре задач (21)-(22). Во второй главе приведено обоснование того, что для них выполняется условие (27). Следовательно, они имеют полиномиальную оценку максимального объема вычислений, необходимых для решения задачи.

Прямой вариант алгоритма (модификация алгоритма С)

Рассматривается параметрическая, линейная по λ , вектор-функция

$$\mathbf{u}^{\mathbf{k}}(\lambda) = \lambda \mathbf{u}^{\mathbf{k}}(1) + (1 - \lambda) \mathbf{u}^{\mathbf{k}}(0).$$

Для ее построения необходимо вычислить два вектора $\mathbf{u}^k(1)$ и $\mathbf{u}^k(0)$, которым в (30) соответствуют значения $\mu = \mu^k$ и $\mu = 0$.

Аналогичным образом задаются вектор-функции $\mathbf{v}^k(\lambda)$ и $\mathbf{w}^k(\lambda)$. Для построения каждой из них также необходимо вычислить по два вектора с компонентами

$$v_j^k(1) = \frac{2\mu^k + (x_j^k)^2 (\mathbf{A}^T \mathbf{u}^k(1))_j}{(x_j^k)^2 + (s_j^k)^2}, \quad v_j^k(0) = \frac{(x_j^k)^2 (\mathbf{A}^T \mathbf{u}^k(0))_j}{(x_j^k)^2 + (s_j^k)^2}, \quad j = 1, \dots, n,$$

$$w_i^k(1) = \frac{2\mu^k - (y_i^k)^2 u_i^k(1)}{(y_i^k)^2 + (q_i^k)^2}, \quad w_i^k(0) = \frac{(y_i^k)^2 u_i^k(0)}{(y_i^k)^2 + (q_i^k)^2}, \quad i = 1, \dots, m.$$

Функции записываются в виде

$$\mathbf{v}^k(\lambda) = \lambda \mathbf{v}^k(1) + (1 - \lambda) \mathbf{v}^k(0),$$

$$\mathbf{w}^k(\lambda) = \lambda \mathbf{w}^k(1) + (1 - \lambda) \mathbf{w}^k(0).$$

Новое значение параметра центрального пути $\mu^{k+1} = \lambda^k \mu^k$ определяется одновременно с новыми значениями переменных двойственной задачи по формулам $\mathbf{u}^{k+1} = \mathbf{u}^k(\lambda^k)$, $\mathbf{v}^{k+1} = \mathbf{v}^k(\lambda^k)$ и $\mathbf{w}^{k+1} = \mathbf{w}^k(\lambda^k)$ до вычисления переменных прямой задачи.

λ^k находим как решение задачи

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \sum_{j=1}^n \left(\lambda \mu^k - x_j^k (\mathbf{v}^k(\lambda) - \mathbf{A}^T \mathbf{u}^k(\lambda))_j \right)^p + \sum_{j=1}^n \left(\lambda \mu^k - s_j^k v_j^k(\lambda) \right)^p +$$

$$+ \sum_{i=1}^m \left(\lambda \mu^k - y_i^k (\mathbf{u}^k(\lambda) + \mathbf{w}^k(\lambda))_i \right)^p + \sum_{i=1}^m \left(\lambda \mu^k - q_i^k w_i^k(\lambda) \right)^p +$$

$$+ \left(\lambda \mu^k - \alpha^k (1 - \mathbf{r}^T \mathbf{u}^k(\lambda)) \right)^p \leq \left(\theta \lambda \mu^k \right)^p, \quad p = 2, 4, \dots$$

Значения переменных прямой задачи вычисляются по формулам (31)-(32) с единственным отличием - в формулы вместо μ^k подставляем значение μ^{k+1} .

Двойственный вариант алгоритма (модификация алгоритма D)

Данный вариант является зеркальным отражением рассмотренного выше прямого варианта. Зафиксируем переменные двойственной задачи \mathbf{u}^k , \mathbf{v}^k и \mathbf{w}^k . Обозначим $\mathbf{g}_1^k = \mathbf{v}^k - \mathbf{A}^T \mathbf{u}^k$, $\mathbf{g}_2^k = \mathbf{v}^k$, $\mathbf{g}_3^k = \mathbf{u}^k + \mathbf{w}^k$, $\mathbf{g}_4^k = \mathbf{w}^k$, $g_5^k = 1 - \mathbf{r}^T \mathbf{u}^k$, а $\mathbf{G}_1^k, \mathbf{G}_2^k, \mathbf{G}_3^k, \mathbf{G}_4^k, \mathbf{G}_5^k$ - соответствующие этим векторам диагональные матрицы. Рассмотрим функцию

$$\begin{aligned} \Phi_2(\mathbf{x}, \mathbf{s}, \mathbf{y}, \mathbf{q}, \alpha, \mathbf{u}^k, \mathbf{v}^k, \mathbf{w}^k, \lambda \mu^k) &= \sum_{j=1}^n \left(\lambda \mu^k - x_j (g_1^k)_j \right)^2 + \sum_{j=1}^n \left(\lambda \mu^k - s_j (g_2^k)_j \right)^2 + \\ &+ \sum_{i=1}^m \left(\lambda \mu^k - y_i (g_3^k)_i \right)^2 + \sum_{i=1}^m \left(\lambda \mu^k - q_i (g_4^k)_i \right)^2 + \left(\lambda \mu^k - \alpha g_5^k \right). \end{aligned}$$

Здесь λ также рассматривается в качестве параметра. Будем минимизировать функцию $\Phi_2(\cdot)$ по переменным прямой задачи при условиях

$$\begin{aligned} \mathbf{A}\mathbf{x} - \mathbf{y} + \alpha \mathbf{r} &= \mathbf{b}, \\ -\mathbf{x} - \mathbf{s} &= -2\mathbf{e}, \\ -\mathbf{y} - \mathbf{q} &= -2\mathbf{e}. \end{aligned} \tag{33}$$

Множителями Лагранжа ограничений-равенств (33) будут векторы $\Delta \mathbf{u}^k(\lambda)$, $\Delta \mathbf{v}^k(\lambda)$ и $\Delta \mathbf{w}^k(\lambda)$. Выполнив ряд преобразований, получим

$$\begin{aligned} \mathbf{x}^k(\lambda) &= \left(\mathbf{G}_1^k \right)^{-2} \left(\mathbf{A}^T \Delta \mathbf{u}^k(\lambda) - \Delta \mathbf{v}^k(\lambda) \right) + \lambda \mu^k \left(\mathbf{G}_1^k \right)^{-1} \mathbf{e}, \\ \mathbf{s}^k(\lambda) &= -\left(\mathbf{G}_2^k \right)^{-2} \Delta \mathbf{v}^k(\lambda) + \lambda \mu^k \left(\mathbf{G}_2^k \right)^{-1} \mathbf{e}, \\ \mathbf{y}^k(\lambda) &= -\left(\mathbf{G}_3^k \right)^{-2} \left(\Delta \mathbf{u}^k(\lambda) + \Delta \mathbf{w}^k(\lambda) \right) + \lambda \mu^k \left(\mathbf{G}_3^k \right)^{-1} \mathbf{e}, \\ \mathbf{q}^k(\lambda) &= -\left(\mathbf{G}_4^k \right)^{-2} \Delta \mathbf{w}^k(\lambda) + \lambda \mu^k \left(\mathbf{G}_4^k \right)^{-1} \mathbf{e}, \\ \alpha^k(\lambda) &= \left(\mathbf{G}_5^k \right)^{-2} \mathbf{r}^T \Delta \mathbf{u}^k(\lambda) + \lambda \mu^k \left(\mathbf{G}_5^k \right)^{-1} \mathbf{e}. \end{aligned} \tag{34}$$

Подставив выражения (34) в равенства (33), получим систему уравнений относительно $\Delta \mathbf{u}^k(\lambda)$, $\Delta \mathbf{v}^k(\lambda)$ и $\Delta \mathbf{w}^k(\lambda)$. Решением ее будут векторы

$$\Delta \mathbf{u}^k(\lambda) = \left(\mathbf{A} \left((\mathbf{G}_1^k)^2 + (\mathbf{G}_2^k)^2 \right)^{-1} \mathbf{A}^T + \left((\mathbf{G}_3^k)^2 + (\mathbf{G}_4^k)^2 \right)^{-1} + \mathbf{r} (\mathbf{G}_5^k)^{-2} \mathbf{r}^T \right)^{-1} \times$$

$$\times \begin{pmatrix} \mathbf{b} + \lambda \mu^k \left(\mathbf{A} (\mathbf{G}_2^k - \mathbf{G}_1^k) - 2 \mathbf{A} (\mathbf{G}_2^k)^2 \right) \left((\mathbf{G}_1^k)^2 + (\mathbf{G}_2^k)^2 \right)^{-1} \mathbf{e} + \\ + \lambda \mu^k \left((\mathbf{G}_3^k - \mathbf{G}_4^k) + 2 \mathbf{G}_4^k \right) \left((\mathbf{G}_3^k)^2 + (\mathbf{G}_4^k)^2 \right)^{-1} \mathbf{e} - \lambda \mu^k (\mathbf{G}_5^k)^{-1} \mathbf{r} \end{pmatrix},$$

$$\Delta v_j^k(\lambda) = \frac{\lambda \mu^k (g_1^k)_j^2 (g_2^k)_j + \lambda \mu^k (g_1^k)_j (g_2^k)_j^2 - 2 (g_1^k)_j^2 (g_2^k)_j^2 - (g_2^k)_j^2 (\mathbf{A}^T \Delta \mathbf{u}^k)_j}{(g_1^k)_j^2 + (g_2^k)_j^2},$$

$$\Delta w_i^k(\lambda) = \frac{\lambda \mu^k (g_3^k)_i^2 (g_4^k)_i + \lambda \mu^k (g_3^k)_i (g_4^k)_i^2 - 2 (g_3^k)_i^2 (g_4^k)_i^2 - (g_4^k)_i^2 \Delta u_i^k}{(g_3^k)_i^2 + (g_4^k)_i^2}.$$

Нетрудно заметить, что все вектор-функции $\Delta \mathbf{u}^k(\lambda)$, $\Delta \mathbf{v}^k(\lambda)$, $\Delta \mathbf{w}^k(\lambda)$, $\mathbf{x}^k(\lambda)$, $\mathbf{s}^k(\lambda)$, $\mathbf{y}^k(\lambda)$, $\mathbf{q}^k(\lambda)$ и $\alpha^k(\lambda)$ являются линейными по параметру λ . В частности, $\mathbf{x}^k(\lambda) = \lambda \mathbf{x}^k(1) + (1 - \lambda) \mathbf{x}^k(0)$, и так далее. Таким образом, для построения каждой из них необходимо вычислить два вспомогательных вектора, соответствующих значениям $\lambda = 1$ и $\lambda = 0$. После этого решаем задачу минимизации λ при условии нахождения в конусе центрального пути

$$\lambda \rightarrow \min_{\lambda \geq 0}, \quad \sum_{j=1}^n \left(\lambda \mu^k - x_j^k(\lambda) (\mathbf{g}_1^k)_j \right)^p + \sum_{j=1}^n \left(\lambda \mu^k - s_j^k(\lambda) (\mathbf{g}_2^k)_j \right)^p +$$

$$+ \sum_{i=1}^m \left(\lambda \mu^k - y_i^k(\lambda) (\mathbf{g}_3^k)_i \right)^p + \sum_{i=1}^m \left(\lambda \mu^k - q_i^k(\lambda) (\mathbf{g}_4^k)_i \right)^p +$$

$$+ \left(\lambda \mu^k - \alpha^k(\lambda) g_5^k \right)^p \leq \left(\theta \lambda \mu^k \right)^p, \quad p = 2, 4, \dots$$

Отыскав значение λ^k , которое соответствует выполнению этого неравенства как строгого равенства, пересчитаем значения параметра центрального пути

$$\mu^{k+1} = \lambda^k \mu^k,$$

переменных прямой задачи

$$\begin{aligned}\mathbf{x}^{k+1} &= \mathbf{x}^k(\lambda^k), \quad \mathbf{s}^{k+1} = \mathbf{s}^k(\lambda^k), \\ \mathbf{y}^{k+1} &= \mathbf{y}^k(\lambda^k), \quad \mathbf{q}^{k+1} = \mathbf{q}^k(\lambda^k), \\ \alpha^{k+1} &= \alpha^k(\lambda^k)\end{aligned}$$

и переменных двойственной задачи

$$\begin{aligned}\mathbf{u}^{k+1} &= \mathbf{u}^k + \frac{1}{\mu^{k+1}} \Delta \mathbf{u}^k(\lambda^k), \\ \mathbf{v}^{k+1} &= \mathbf{v}^k + \frac{1}{\mu^{k+1}} \Delta \mathbf{v}^k(\lambda^k), \\ \mathbf{w}^{k+1} &= \mathbf{w}^k + \frac{1}{\mu^{k+1}} \Delta \mathbf{w}^k(\lambda^k).\end{aligned}$$

Другой интерпретацией двойственного алгоритма может являться решение методом Ньютона задачи максимизации двойственной логарифмической барьерной функции

$$\begin{aligned}\mathbf{b}^T \mathbf{u} - 2 \sum_{j=1}^n v_j - 2 \sum_{i=1}^m w_i + \\ + \mu \sum_{j=1}^n \ln(g_1)_j + \mu \sum_{j=1}^n \ln(g_2)_j + \mu \sum_{i=1}^m \ln(g_3)_i + \mu \sum_{i=1}^m \ln(g_4)_i + \mu \ln g_5 \rightarrow \max\end{aligned}$$

при условиях

$$\mathbf{g}_1^k = \mathbf{v}^k - \mathbf{A}^T \mathbf{u}^k, \quad \mathbf{g}_2^k = \mathbf{v}^k, \quad \mathbf{g}_3^k = \mathbf{u}^k + \mathbf{w}^k, \quad \mathbf{g}_4^k = \mathbf{w}^k, \quad g_5^k = 1 - \mathbf{r}^T \mathbf{u}^k.$$

Во второй главе продемонстрировано, что обе интерпретации приводят к одним и тем же расчетным формулам.

Можно объединить идеи прямого и двойственного алгоритма в одном прямо-двойственном алгоритме (модификации алгоритма Е). Каждая итерация состоит из двух шагов: прямого и двойственного. На прямом шаге закрепляются переменные прямой задачи, а переменные двойственной задачи представляются в виде параметрических функций, после чего в максимальной степени уменьшается значение параметра центрального пути μ . На двойственном шаге, напротив, закрепляются переменные двойственной

задачи, переменные прямой задачи представляются в виде параметрических функций, и снова максимально возможно уменьшается значение μ .

Завершение итеративного процесса

Так же, как и в рассмотренных в предыдущем разделе аффинно-масштабирующих алгоритмах, в вычислительную схему включается следующий пункт. Если на некоторой итерации выполняется условие

$$\phi(\mathbf{u}^k) = \mathbf{b}^T \mathbf{u}^k + 2\mathbf{e}^T \mathbf{u}^k - 2\mathbf{e}^T (\mathbf{A}^T \mathbf{u}^k)_+ > 0,$$

то система (20) несовместна.

Завершение вычислительного процесса в случае совместности ограничений также формализуется. На каждой итерации проверяется, нельзя ли продвинуться вдоль направления корректировки прямых переменных с шагом $\gamma^k = -\alpha^k / \Delta\alpha^k$, то есть нарушается ли хоть одно ограничение-неравенство в точке

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \gamma^k \Delta\mathbf{x}^k, & \mathbf{s}^{k+1} &= \mathbf{s}^k + \gamma^k \Delta\mathbf{s}^k, \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + \gamma^k \Delta\mathbf{y}^k, & \mathbf{q}^{k+1} &= \mathbf{q}^k + \gamma^k \Delta\mathbf{q}^k. \end{aligned}$$

Если все они выполняются, то имеем допустимую точку системы (20).

Экспериментальное исследование алгоритмов

Было осуществлено экспериментальное исследование предложенных алгоритмов. В частности, на 14 несовместных задачах проверялась работоспособность приведенного критерия несовместности системы (20). Критерий идентифицировал несовместность каждой из них на первой же итерации. Также алгоритмы были протестированы на 14 совместных задачах размерности от 30×80 до 41×80 , полученных из моделей управления режимами ЭЭС. В целях преемственности были выбраны те же задачи, на которых исследовались варианты аффинно-масштабирующих алгоритмов.

Приведем таблицу, в которую занесем число итераций, потребовавшееся алгоритмам для получения решения каждой из них. Также для сопоставления приведем результаты, показанные на этих задачах рассмотренным ранее вариантом F_1 аффинно-масштабирующего метода, используемом в данное время на практике, в частности, в программно-вычислительном комплексе СДО. Число итераций может служить сравнительной характеристикой алгоритмов, поскольку для всех алгоритмов, кроме прямо-двойственных, наибольшую сложность представляет обращение симметричной положительно определенной матрицы размерности $m \times m$. Напомним, что сложность одной итерации прямо-двойственных алгоритмов, использующих 2 обращения матрицы, соответственно в 2 раза больше.

Таблица 5. Число итераций, необходимое для решения совместных задач

Алгоритмы	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Средн.
C	7	6	7	10	5	24	26	21	22	23	23	20	21	21	16.9
C₄	5	5	5	7	3	14	14	12	13	13	13	12	12	13	10.1
C₁₆	4	4	5	5	2	10	11	10	10	11	10	10	10	10	8
E₄	4	3	4	5	3	10	10	9	10	10	10	9	9	9	7.5
E₁₆	3	2	3	3	2	6	6	6	6	6	6	6	6	6	4.8
F₁	4	8	7	8	3	11	13	11	11	12	12	11	11	11	9.5

Из таблицы видно, что предложенные алгоритмы центрального пути, обладая полиномиальными оценками, оказываются эффективными и на практике, что обусловлено, в частности, наличием стартовой точки, принадлежащей центральному пути, эффективными процедурами завершения вычислительного процесса (и то, и другое отсутствует при решении задач линейного программирования), а также существенным эффектом применения параметризаций. Как и на паре взаимно-двойственных задач линейного программирования, повышение степеней p при решении вспомогательных задач приводит к улучшению результатов, хотя теоретическое обоснование алгоритмов с $p > 4$ отсутствует.

Некоторые направления дальнейших исследований

Недостатком алгоритмов центрального пути при решении систем нелинейных уравнений и неравенств на основе итеративной линеаризации является то, что вычислительный процесс начинается с автоматически сформированной стартовой точки $\mathbf{x}^1 = \mathbf{e}$, соответствующей середине допустимого интервала для системы (5). В то же время перспективнее использовать в качестве стартового приближения точку \mathbf{x}^t , полученную с предыдущей итерации линеаризации. Здесь могут помочь адаптированные к системам линейных уравнений и неравенств алгоритмы оптимизации в конусе скошенного пути, поскольку они могут стартовать с любой относительно внутренней точки множества допустимых решений, в частности, и с интересующей нас

$$x_j^1 = 2 \frac{x_j^t - \underline{x}_j}{\bar{x}_j - \underline{x}_j}, \quad s_j^1 = 2 - x_j^1, \quad j = 1, \dots, n, \quad y_i^1 = q_i^1 = 1, \quad i = 1, \dots, m, \quad \alpha^1 = 1, \\ u_i^1 = 0, \quad i = 1, \dots, m, \quad v_j^1 = 1, \quad j = 1, \dots, n, \quad w_i^1 = 1, \quad i = 1, \dots, m.$$

Алгоритмы скошенного пути адаптируются к системам линейных уравнений и неравенств аналогично алгоритмам центрального пути. Начальный коэффициент скошенности составляет

$$\gamma = \sum_{j=1}^n x_j^1 / n \min_{j=1, \dots, n} x_j^1.$$

Комбинированный алгоритм

Другим перспективным способом решения системы (5) может служить следующий алгоритм, вычислительный процесс в котором начинается с произвольной пары векторов, удовлетворяющей в строгой форме ограничениям-неравенствам

$$\bar{\mathbf{x}} > \mathbf{x}^1 > \underline{\mathbf{x}}, \quad \bar{\mathbf{y}} > \mathbf{y}^1 > \underline{\mathbf{y}}.$$

На каждой итерации вычисляем вектор невязки ограничений-равенств

$$\mathbf{r}^k = \mathbf{y}^k - \mathbf{A}\mathbf{x}^k.$$

Направление корректировки $(\Delta \mathbf{x}^k(\lambda_x, \lambda_y), \Delta \mathbf{y}^k(\lambda_x, \lambda_y))$ ищем как решение задачи

$$\begin{aligned} & -\lambda_x \left((\mathbf{D}_k^1)^{-1} - (\mathbf{D}_k^2)^{-1} \right) \Delta \mathbf{x} + \frac{1}{2} (\Delta \mathbf{x})^T \left((\mathbf{D}_k^1)^{-2} + (\mathbf{D}_k^2)^{-2} \right) \Delta \mathbf{x} - \\ & -\lambda_y \left((\mathbf{D}_k^3)^{-1} - (\mathbf{D}_k^4)^{-1} \right) \Delta \mathbf{y} + \frac{1}{2} (\Delta \mathbf{y})^T \left((\mathbf{D}_k^3)^{-2} + (\mathbf{D}_k^4)^{-2} \right) \Delta \mathbf{y} \rightarrow \min_{\Delta \mathbf{x}, \Delta \mathbf{y}}, \end{aligned} \quad (35)$$

$$\mathbf{A} \Delta \mathbf{x} - \Delta \mathbf{y} = \mathbf{r}^k. \quad (36)$$

Здесь

$$\mathbf{D}_k^1 = \text{diag}\{x_j^k - \underline{x}_j\}, \mathbf{D}_k^2 = \text{diag}\{\bar{x}_j - x_j^k\}, \mathbf{D}_k^3 = \text{diag}\{y_i^k - \underline{y}_i\}, \mathbf{D}_k^4 = \text{diag}\{\bar{y}_i - y_i^k\}.$$

Пара значений $\lambda_x = 0$, $\lambda_y = 0$ соответствует варианту аффинно-масштабирующего метода. При $\lambda_x = 1$, $\lambda_y = 1$ целевая функция (35) является квадратичной аппроксимацией логарифмической барьерной штрафной функции, что делает алгоритм идейно сходным с алгоритмами центрального пути. При этом, в отличие от них, здесь не требуется начального приближения, принадлежащего конусу центрального пути. В целом, чем больше величины λ_x и λ_y , тем ярче выражено центрирующее направление.

Решением задачи (35), (36) будут векторы

$$\begin{aligned} \Delta \mathbf{x}^k(\lambda_x, \lambda_y) &= \left((\mathbf{D}_k^1)^{-2} + (\mathbf{D}_k^2)^{-2} \right)^{-1} \left(\mathbf{A}^T \mathbf{u}^k(\lambda_x, \lambda_y) + \lambda_x (\mathbf{D}_k^1)^{-1} \mathbf{e} - \lambda_x (\mathbf{D}_k^2)^{-1} \mathbf{e} \right), \\ \Delta \mathbf{y}^k(\lambda_x, \lambda_y) &= \left((\mathbf{D}_k^3)^{-2} + (\mathbf{D}_k^4)^{-2} \right)^{-1} \left(-\mathbf{u}^k(\lambda_x, \lambda_y) + \lambda_x (\mathbf{D}_k^3)^{-1} \mathbf{e} - \lambda_x (\mathbf{D}_k^4)^{-1} \mathbf{e} \right), \end{aligned}$$

где вектор множителей Лагранжа ограничений (36) выражается в виде

$$\begin{aligned} \mathbf{u}^k(\lambda_x, \lambda_y) &= \left(\mathbf{A} \mathbf{Z}_x^k \mathbf{A}^T + \mathbf{Z}_y^k \right)^{-1} \times \\ &\times \left(\mathbf{r}^k - \mathbf{A} \mathbf{Z}_x^k (\lambda_x \mathbf{D}_k^1 \mathbf{e} - \lambda_x \mathbf{D}_k^2 \mathbf{e}) + \mathbf{Z}_y^k (\lambda_y \mathbf{D}_k^3 \mathbf{e} - \lambda_y \mathbf{D}_k^4 \mathbf{e}) \right) \end{aligned}$$

при

$$\mathbf{Z}_x^k = \left((\mathbf{D}_k^1)^{-2} + (\mathbf{D}_k^2)^{-2} \right)^{-1}, \quad \mathbf{Z}_y^k = \left((\mathbf{D}_k^3)^{-2} + (\mathbf{D}_k^4)^{-2} \right)^{-1}.$$

Поскольку $\Delta \mathbf{x}^k$ и $\Delta \mathbf{y}^k$ представляют собой линейные относительно λ_x и

λ_y вектор-функции, то можно поставить задачу выбора λ_x и λ_y , доставляющих максимально возможный (из условий невыхода переменных

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \gamma^k \Delta \mathbf{x}^k, \quad \mathbf{y}^{k+1} = \mathbf{y}^k + \gamma^k \Delta \mathbf{y}^k.$$

за границы допустимых интервалов) шаг корректировки γ^k . Как и ранее, шаг $\gamma^k = 1$ соответствует нахождению допустимого решения системы (5).

Вектор \mathbf{u}^k множителей Лагранжа ограничений (36) также может использоваться для идентификации несовместности на основе критерия (10) при любых значениях λ_x и λ_y .

Как показало экспериментальное исследование, данный комбинированный алгоритм успешно решает линейризованные задачи. Кроме того, как правило, результатом его работы является точка, находящаяся ближе к середине допустимого интервала, что является полезным свойством при решении исходной нелинейной задачи. Данные экспериментального исследования, проведенного на тех же 14 совместных задачах, приведем в таблице 6.

Таблица 6. Число итераций, необходимое для решения совместных задач комбинированным алгоритмом

Задачи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Среднее
Число итераций	2	4	4	4	1	8	8	7	7	8	7	7	6	7	5.7

Интересно сопоставить аффинно-масштабирующий и комбинированный алгоритмы на исходной нелинейной задаче поиска допустимых режимов ЭЭС. Для сопоставления выберем 4 реальные схемы, расчеты на которых осуществлялись О.Н.Войтовым, - ZMIN (152 узла), IEEE118F (118 узлов), KUZBF (207 узлов) и ZIMKRASN (211 узлов). В таблице 7 приведем суммарное число итераций (равное числу обращений матрицы), необходимых для получения допустимого режима, а также число, потребовавшихся для этого итераций линейризации.

**Таблица 7. Число итераций, необходимое для решения
исходной задачи поиска допустимых режимов ЭЭС**

Алгоритмы / схемы	ZMIN	IEEE118F	KUZBF	ZIMKRASN
Афф.-масшт. (F1)	11 (3глоб.)	54 (11глоб.)	39 (7глоб.)	74 (9глоб.)
Комбинированный	4 (3глоб.)	50 (17глоб.)	33 (3глоб.)	55 (6глоб.)

Перспективным представляется использование комбинированного алгоритма и для решения задач линейного программирования. Значения λ_x и λ_y в этом случае могут выбираться из условия максимального уменьшения целевой функции задачи (1.1). Изучение и теоретическое обоснование такого алгоритма является предметом дальнейших исследований.

Важным направлением дальнейших исследований, в результате которых может быть достигнуто значительное ускорение работы алгоритмов, связано с альтернативными способами решения вспомогательной задачи. Вместо точного обращения матрицы на каждой итерации, можно учесть, что по итерациям изменяется только диагональная матрица весовых коэффициентов $\mathbf{D}_k = \text{diag}\{d_j^k\}$, и эти изменения не столь велики. В частности, Кармаркаром в [51] была предложена техника “частичного обновления”, позволяющая уменьшить теоретическую сложность алгоритмов внутренних точек в \sqrt{n} раз.

Для упрощения предположим, что на некоторой итерации матрица \mathbf{D}_k была единичной, а на следующей - стала равной \mathbf{D} . Обозначим j -столбец матрицы \mathbf{A} за \mathbf{a}_j . Тогда

$$\mathbf{A}\mathbf{D}\mathbf{A}^T = \sum_{j=1}^n d_j \mathbf{a}_j \mathbf{a}_j^T = \sum_{j=1}^n \mathbf{a}_j \mathbf{a}_j^T + \sum_{j=1}^n (d_j - 1) \mathbf{a}_j \mathbf{a}_j^T = \mathbf{A}\mathbf{A}^T + \sum_{j=1}^n (d_j - 1) \mathbf{a}_j \mathbf{a}_j^T.$$

Рассмотрим гипотетическую ситуацию, когда $d_j = 1$ для всех $j \neq 1$. В этом случае $\mathbf{A}\mathbf{D}\mathbf{A}^T$ представляет собой одноранговую модификацию $\mathbf{A}\mathbf{A}^T$:

$$\mathbf{A}\mathbf{D}\mathbf{A}^T = \mathbf{A}\mathbf{A}^T + (d_1 - 1) \mathbf{a}_1 \mathbf{a}_1^T.$$

По формуле Шермана-Моррисона получаем

$$\left(\mathbf{A}\mathbf{D}\mathbf{A}^T\right)^{-1} = \left(\mathbf{A}\mathbf{A}^T\right)^{-1} - (d_1 - 1) \frac{\left(\mathbf{A}\mathbf{A}^T\right)^{-1} \mathbf{a}_1 \mathbf{a}_1^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1}}{1 + (d_1 - 1) \mathbf{a}_1^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1} \mathbf{a}_1}.$$

Данное преобразование требует порядка $O(n^2)$ арифметических операций.

В общем случае, когда все компоненты d_j отличны от единицы, обращение матрицы $\mathbf{A}\mathbf{D}\mathbf{A}^T$ потребует n подобных одноранговых преобразований, то есть сложность одной итерации по-прежнему составляет $O(n^3)$.

Идея техники частичного обновления состоит в том, что производятся только одноранговые преобразования, соответствующие компонентам j , для которых величина $|d_j - 1|$ превосходит некоторое пороговое значение. Фактически на каждой итерации матрица весовых коэффициентов \mathbf{D}_k заменяется на ее аппроксимацию $\tilde{\mathbf{D}}_k$.

Анализ алгоритмов внутренних точек, использующих технику частичного обновления должен состоять из двух дополнительных процедур. Во-первых, необходимо показать, что при использовании модифицированной матрицы весовых коэффициентов алгоритм сохраняет свойства исходного алгоритма, в частности, полиномиальность. Второй желательной процедурой является подсчет максимального количества необходимых одноранговых преобразований.

Подробное исследование вычислительных аспектов техники частичного обновления провел Д.Шанно в работе [66]. К сожалению, несмотря на серьезное сокращение оценки максимального объема вычислений, применение частичного обновления на практике, как показали экспериментальные исследования, не дает пока существенных положительных результатов. В то же время вспомогательная задача и ее приближенное решение по-прежнему остаются главными из направлений исследований, благодаря которым в будущем можно ожидать улучшения работы алгоритмов внутренних точек.