
БИЗНЕС-ИНФОРМАТИКА

УДК 620.9:662.6:004.8

ОПТИМИЗАТОР ПОДСИСТЕМЫ ДИАГНОСТИКИ СОСТОЯНИЯ ЭНЕРГОУСТАНОВОК, SKAIS, ДЛЯ РЕШЕНИЯ ЗАДАЧ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

Г.Д. Крохин, А.Е. Некипелов

Новосибирский государственный университет
экономики и управления «НИНХ»
E-mail: g.d.krokhin@nsuem.ru

Диагностический контроль энергоустановок тепловой электростанции (ТЭС) осуществляется с помощью экспертной диагностической системы функционально-гибридного типа с именем SKAIS. SKAIS – «Система контроля, анализа и слежения за изменением состояния энергоустановки» – управляемый в диалоговом режиме программный комплекс, ориентированный на диагностирование и экспертизу энергоустановок любых типов. SKAIS позволяет на ранней стадии (с использованием экспресс-испытаний) диагностировать снижение экономичности, определять величину, причины и опасность происходящих изменений, прогнозировать состояние, оценивать надежность, остаточный ресурс, долговечность, степень риска и ущерб от продолжения дальнейшей эксплуатации энергоустановки. Система SKAIS осуществляет принятие решений на выход из создавшейся конфликтной ситуации (вывод в ремонт или введение ограничения) с представлением подготовленных в базе знаний рекомендаций (в виде готовых продукций – решающих правил) оперативному и ремонтному персоналу тепловой электростанции. Для этого создается база данных и знаний (БД и З) обо всех вынужденных остановках и дефектах оборудования, отклонениях от правил его нормальной эксплуатации.

Ключевые слова: комбинационная оптимизация, четкая и нечеткая постановки решения, SKAIS, глобальный минимум функции цели, диагностика и экспертиза.

OPTIMIZER OF A DIAGNOSTIC SUBSYSTEM OF POWER INSTALLATIONS (SKAIS) STATE FOR SOLUTION OF PROBLEMS OF MAINTENANCE SERVICE

G.D. Krokhin, A.Ye. Nekipelov

Novosibirsk State University of Economy and Management
E-mail: g.d.krokhin@nsuem.ru

The paper considers the features of the diagnostic control of power installations of power plant via expert diagnostic system of functional hybrid type named SKAIS. SKAIS, the system of control, analysis and monitoring of changes in the power plant state, is a dialogue mode-driven software system, designed for diagnostics and expertise of any power plants. SKAIS makes it possible to diagnose decline in economical efficiency, define amount, reasons and

danger of current changes, forecast state, evaluate reliability, remaining life, longevity, risk level and damage from further power installation exploitation at early stages (with the use of accelerated tests). SKAIS system makes decisions toward surmounting the conflict (maintenance or restriction) with the presentation of recommendations, prepared in the knowledge base (in the form of end products – decision rules) to operative and maintenance personnel of a thermal plant. The data base and knowledge base about all forced stops and equipment defects, deviations from the exploitation norms are made for that end.

Key words: combinational optimization, well-defined and ill-defined statement of solution, SKAIS, global minimum of objective function, diagnostics and expertise.

Введение. В интеллектуальном центре подсистемы контроля и диагностики энергоустановок ТЭС – SKAIS (рис. 1, 2, см. Приложение) работает программный модуль OPTIMIZATOR по схеме MIN + POISK N (рис. 3, 4, см. Приложение). Основное назначение модуля – программное решение задач минимизации функций цели, имеющих несколько минимумов, но достаточно гладких в окрестностях каждого из них. Вдали от минимумов допускаются неустранимые разрывы первого рода. Для нахождения окрестности глобального минимума используется из MIN метод случайного поиска (его комбинация). Методом сопряженных градиентов минимум уточняется. При появлении «оврагов» градиентные методы отказываются работать. В этом случае подключается к POISK N овражный метод И.М. Гельфанда, который позволяет осуществить многомерный поиск минимума [7, 8]. Из точки P_0 по двум направлениям выполняется наискорейший спуск на дно оврага (рис. 6, см. Приложение). Для этого при вычислении градиента вначале берем шаг $h_1 = h$, а потом шаг $h_2 = h/10$, получаем точки Q_1 и Q_2 , определяющие прямую линию – «дно оврага». По «дну оврага» выполняем еще один многомерный поиск минимума. Получим точку P_1 . При необходимости этот элемент поиска можно повторить. Последовательность P_0, P_1, \dots задает убывающую последовательность целевой функции $F(P_0) > F(P_1) \dots$. После срабатывания правила остановки, когда реализуется заданный порог схожести $POX(P_0, P_1) \cong 1$ (рис. 5 и ПП с именем SIM, см. Приложение), последняя полученная точка \bar{x}^* и значение функции в этой точке будут точкой глобального минимума.

Результаты анализа сравниваемых состояний представляются визуально, в виде их функциональных значений $POX = P(X, Y)$ на экране монитора (см. рис. 5, табл. П.1).

Похожесть диагностируемых состояний определяется с помощью программы POX из ВК SKAIS. В геометрической интерпретации значения X отображают (представляют) собой колебательную линию, построенную по параметрам оцениваемого состояния относительно его эталонного значения Y .

Порог схожести задается разработчиком OPTIMIZATORa. Точки Q_1 и Q_2 считаются лежащими на дне оврага. Эти две точки определяют прямую линию, по которой осуществляется одномерный поиск минимума. Одно из направлений в точке P_0 является градиентным. Второе направление будет случайным. Эта комбинация детерминированного и случайного поиска приводит к желаемому результату. Регулирующими параметрами элемента поиска является пара (R, h) , где R – первоначальный шаг. В качестве правила остановки, при наискорейшем спуске на дно оврага и принятом механизме случайного выбора решения из полученного набора эвристик используется принцип «по-

хожести» точек [4, 5, 9]. Одной из эвристик алгоритма является предварительное знание об области D и примерной зоне поиска x^* , в которой находится минимум функции $y = f(x)$, а также знание об изменении параметров технологического процесса по их «похожести» (значения РОХ должны быть одинаковыми или близкими). Эта информация позволит оценить первоначальный шаг R и значения функции $y = f(x)$ в выбранных точках.

Регулирование этими параметрами с использованием «меры схожести» РОХ позволяет эффективно (и визуально на экране монитора) применять диалоговый подход при оптимизации задач тепловой электростанции (ТЭС). В этом случае лицу, принимающему решение (ЛПР) желательно иметь не одно, а группу хороших решений и возможности принятия решения по времени *on-line* для последующего выбора окончательного оптимального решения (например, при базовом и переходном режимах энергоустановки). Для реализации этого в используемые эвристические правила вводится неопределенность исходной информации, благодаря чему и будет порождаться класс субоптимальных решений. Эвристические правила, обладающие ограниченной неопределенностью, назовем «размытыми» эвристиками [1, 4, 9]. Результатом поиска будет единственное решение, близость которого к оптимальному решению определяется величиной максимальной схожести РОХ, если мы имеем дело с тестовой задачей.

Тестовые задачи сконструированы автором работы так, чтобы выделить особенности теплоэнергетического процесса [5, 8]. При опробовании OPTIMIZATORa по схеме MIN + POISK N на тестовых примерах (использованы функции Х.Х. Розенброка $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, Л.А. Растринина $Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$) алгоритм сходится, т.е. минимум осуществляется из любой начальной точки. Использование механизма случайного выбора решений позволяет расширять область возможных реализаций или сужать ее, в зависимости от ситуации и готовности энергоустановки по состоянию к выполнению режима.

Задачи математического программирования при четкой постановке в приложении к задачам технической диагностики энергоустановок ТЭС решаем следующим образом.

Пусть в области R , определяемой ограничениями

$$\bar{a} \leq x \leq \bar{b}, t_j(x) \leq b_j, j = 1, 2, \dots, m, \quad (1)$$

задана целевая, в общем случае, нелинейная функция $t(x)$. Требуется найти такой $\bar{x} \in R$, для которого справедливо

$$t(\bar{x}) = \min_{x \in R} t(x). \quad (2)$$

Здесь условие (1) означает, что каждая компонента $t_j(x)$, $j = 1, 2, \dots, m$ вектора (матрицы) \bar{x} изменяется в пределах от соответствующего наименьшего допустимого значения a до наибольшего допустимого значения b . Количество компонент – n . Любой вектор $\bar{x} \in R$ называется допустимым. Вектор $\bar{x} \in R$ назовем оптимальным, если для любого другого вектора $x \in R$ выполняется условие:

$$t(\bar{x}) \leq t(x). \quad (3)$$

Как известно из теории исследования операций [10], использование градиентных методов для решения многоэкстремальных задач затруднительно и малоэффективно, так как необходимы полные исследования целевой функции (определение, можно приближенное, вида поверхности, начальные приближения). Поэтому, как это подтверждает практика, наиболее результативными методами поиска минимума могут быть различные модификации случайного поиска. С этой целью в модуль OPTIMIZATOR включена модификация метода случайного поиска – метод Монте-Карло [5, 10].

Формула изменения координат вектора \bar{x} имеет следующий вид:

$$x_i = a_i + \xi(b_i - a_i), \quad (4)$$

где ξ – случайное число, $0 \leq \xi \leq 1$;

a_i, b_i – нижние и верхние ограничения на переменные.

Из (4) следует, что точка X_i при любом ξ всегда находится в области R (в ограничителях). Для определения случайного числа ξ используется процедура RAND, вырабатывающая случайные числа, необходимые при поиске окрестности глобального минимума функции цели. Попадание в окрестность глобального минимума происходит с некоторой вероятностью:

$$p \cong 1 - (1 - \varepsilon/\rho)^r, \quad (5)$$

где ε – объем зоны критерия глобального минимума;

ρ – объем зоны поиска;

r – количество случайных бросков.

Работа метода прекращается, если количество случайных бросков превышает заданное целое число k . Из совокупности точек, полученных в результате случайных бросков, выбирается точка X^0 , которая соответствует наименьшему значению функции цели $t(x)$. Однако метод Монте-Карло нецелесообразно использовать для нахождения $t(\bar{x})$, так как вероятность случайного попадания в ε -окрестность на одном шаге поиска, определяемая отношением объемов n -мерных гиперсфер с радиусами ε и ρ (начальным расстоянием до цели), равна:

$$p = (\varepsilon/\rho)^n. \quad (6)$$

Среднее число шагов, необходимое для случайного попадания в ε -окрестность цели

$$N_{\text{ср}} = (\rho/\varepsilon)^n, \quad (7)$$

имеет экспоненциальный характер и, следовательно, с ростом n быстро растет и $N_{\text{ср}}$. В схеме MIN + POISK N этот метод используется для двух целей:

- 1) проведения статистических испытаний и расчетных экспериментов на ЭВМ;
- 2) оценки окрестности глобального экстремума функции цели.

В последнем случае точка X^0 , найденная методом Монте-Карло, улучшается постепенным приближением к цели путем ограничения поиска зоной, стягивающейся к наилучшей случайной пробе. Это значит, что случайные пробы производятся в объеме, центром которого является точка с наимень-

шим значением функции цели. По мере производства случайных испытаний этот объем стягивается к своему центру.

Если в процессе испытаний была найдена точка с меньшим значением функции цели, то объем испытаний устанавливается вокруг этой новой точки. Таким образом, зона испытаний перемещается в район цели, причем на каждом шаге вероятность случайного нахождения наилучшей точки становится все большей. Этот принцип лежит в основе второй модификации случайного поиска – методе случайного направленного поиска [2, 5, 10].

Из точки X^0 делается случайный шаг

$$\Delta X^0 = (2\xi - 1)c_i/f, \quad (8)$$

$$\text{где } c_i \cong (\text{abs } b_i - \text{abs } a_i)/2. \quad (9)$$

На начальном шаге принимается $f = 1$. Затем определяется координата новой точки

$$X = X^0 + \Delta X^0 \quad (10)$$

и сравниваются значения

$$t(X^0 + \Delta X^0) \geq t(X^0). \quad (11)$$

При этом $t(X^0) \leq t(X^0 + \Delta X^0)$ считается неудачной ситуацией, а при $t(X^0) > t(X^0 + \Delta X^0)$ проверяется условие

$$abc \{ [t(x^0) - t(x^0 + \Delta x^0)] / t(x^0 + \Delta x^0) \} \leq \delta, \quad (12)$$

при выполнении которого ситуацию также считают неудачной и удачной – в противном случае. При неудаче предусмотрен возврат в точку X^0 , из которой делается шаг в диаметрально противоположном направлении с последующей проверкой условия (12). В случае неудачи вновь происходит возвращение в точку X^0 , из которой делается столько случайных шагов, сколько потребуется для нахождения удачной ситуации. Если такая точка найдена, то через нее и X^0 проводится вектор, в направлении которого начинается движение с постоянным шагом.

При движении по выбранному направлению проверяется относительное изменение функции цели

$$d = abc \{ [t(x^k) - t(x^{k+1})] / t(x^{k+1}) \}. \quad (13)$$

В случае $d \leq \delta$ предусмотрено возвращение в точку X^k с последующим выбором (с помощью случайных испытаний) нового направления.

Значение f меняется в процессе минимизации следующим образом. Как только число неудачных шагов фиксированной точки окажется равным заданному целому числу K , то f увеличивается на единицу.

Эта операция позволяет осуществлять поиск и движение в выбранном направлении с все более и более уменьшающимися шагами.

Метод случайного направленного поиска прекращает свою работу, если выполняется условие $f = \omega$ (заданное целое число $\gg 1$). Если координаты точ-

ки вдруг оказываются вне ограничений, то функции присваивается число 10^{10} (или любое другое, определяемое на основе расчетных экспериментов).

Предполагая, что окрестность глобального минимума найдена, продолжается дальнейшее улучшение точки минимума. Для этого в схеме MIN + POISK N используется метод сопряженных градиентов с преобразованием координат [2, 5, 10] (см. рис. 3).

Задачи математического программирования в нечетких условиях в приложении к задачам технической диагностики энергоустановок решаются следующим образом [3, 5].

Под ситуацией принятия решений при выборе диагноза состояния энергоустановки условимся понимать:

- множество альтернатив, из которых лицо, принимающее решение (ЛПР), производит выбор;
- множество ограничений, накладываемых на этот выбор;
- целевую функцию, которая позволяет ЛПР ранжировать имеющиеся у него альтернативы.

В результате каждое ЛПР, имея множество сформулированных целей, способно определить свои предпочтения.

Но на практике, особенно при диагностике в реальном масштабе времени, картина принятия решений резко меняется, так как ЛПР вынуждено применять следующее утверждение: « Z должно быть в окрестности Y »! А это уже подчеркивает появление нечеткости в формулировании цели, согласно [1, 3, 6].

Выражение «в окрестности...» представим нечетким подмножеством X , определяемым функцией (точнее, ее отображением f):

$$f: X \rightarrow L, \quad (14)$$

где L – полная дистрибутивная решетка; X – множество альтернатив.

В результате принятия решения по одному из предлагаемых диагнозов представим возникшую нечеткую обстановку как множество X -альтернатив вместе с его нечеткими подмножествами. Эти подмножества представляют собой также нечетко сформулированные критерии (цели и ограничения), т.е. систему:

$$(X, f_0, \dots, f_n, L). \quad (15)$$

Здесь f_0, \dots, f_n – целевые функции.

Перебирая, по возможности, все критерии при выборе наиболее предпочтительного диагноза, можно построить функцию

$$D = f_0 \vee \dots \vee f_n. \quad (16)$$

Оптимум в этом случае будет соответствовать той области X , элементы которой максимизируют диагноз D . В результате проведенных рассуждений можно определить нечеткую обстановку такой задачи тройкой (X, D, L) . Предположим при этом, что решение задачи диагноза в нечеткой постановке будет определяться в виде нечеткого подмножества универсального множе-

ства альтернатив. Под оптимальным решением при этом будем понимать элемент $x_0 \in X$ (если такой существует), для которого, согласно [3, 11]:

$$D(x_0) = \sup_{x \in X} D(x), \quad (17)$$

где $D = f_0 \vee \dots \vee f_n$ – нечеткое решение; \sup означает \cup , \inf означает \cap . Таким образом, задача минимизации решений при выборе предположительного диагноза сведена к задаче нечеткого математического программирования (НМП), т.е. к задаче многокритериальной оптимизации.

Итак, под задачей НМП будем понимать задачу нахождения $\sup_{f_1 \cup \dots \cup f_n} f_0(x)$, или

$$\sup_{x \in X} [f_0(x) \vee f_1(x) \vee \dots \vee f_n(x)] = \sup_{x \in X} D(x). \quad (18)$$

Результат решения задачи минимизации функции цели

В данной работе было уделено внимание на реализации комплексного решения оптимального нахождения минимума различных функций цели с помощью среды MathCAD 14. Рассмотрены в комплексе метод Монте-Карло и градиентный метод наискорейшего спуска, позволяющий уточнить минимум функции, начальное приближение которого получается из случайного поиска методом Монте-Карло.

1. Метод Монте-Карло

Для минимизации функции многих переменных разработано множество численных методов, но большинство из них связано с подсчетом градиента функции, что со своей стороны может дать эффективные алгоритмы вычисления, если удастся аналитически подсчитать частные производные. Между тем более универсальным методом минимизации функции многих переменных является метод перебора, при котором произвольным образом разбивается область определения функций на симплексы и в каждом узле симплекса вычисляется значение функции цели, причем происходит сравнение – перебор значений и на печать выводится точка минимума и значение функции в этой точке.

В методе Монте-Карло зададим функцию $z = f(x_1, x_2, \dots, x_n)$. Выбираем область поиска решения задачи:

$$a_i \leq x_i \leq b_i, \quad i = \overline{1, n};$$

а) производим случайные броски, т.е. выбираем значения $x_i^{(m)}$ для каждой переменной x_i по формуле

$$x_i^{(m)} = a_i + \text{random}(b_i - a_i), \quad \text{где } i = \overline{1, n}, \quad m \in N;$$

б) сравниваем значения функции:

$$z^{(m)} = f(x_1^{(m)}, x_2^{(m)}, \dots, x_n^{(m)}) \leq z^{(m+1)} = f(x_1^{(m+1)}, x_2^{(m+1)}, \dots, x_n^{(m+1)}),$$

если это неравенство выполняется, то $z_{\min} = z^{(m)}$, если не выполняется, то $z_{\min} = z^{(m+1)}$;

в) количество случайных бросков либо фиксировано, либо уточняется при достижении необходимой определенной погрешности.

Здесь m – число итераций. Процесс итерации останавливается, когда достигается требуемая предельная погрешность, т.е. когда выполнены условия остановки итерации:

$$\max_{i=1} |x_i^{m+1} - x_i^m| < \varepsilon, \quad i = \overline{1, n}.$$

Данный метод хорош лишь в том случае, когда имеется некое первое оптимальное приближение, в противном случае результат может быть совсем иной, не удовлетворяющий критериям. Поэтому в данной работе мы используем его в комплексе с методом Монте-Карло, в дальнейшем, получив начальное приближение, уточним его до некоторой погрешности градиентным методом наискорейшего спуска.

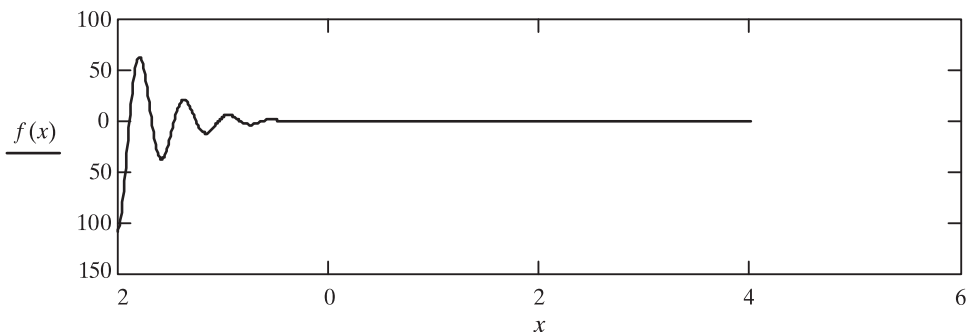
3. Пример реализации и решения задачи определения минимума функции цели в среде MATHCAD

Пусть задана многоэкстремальная функция

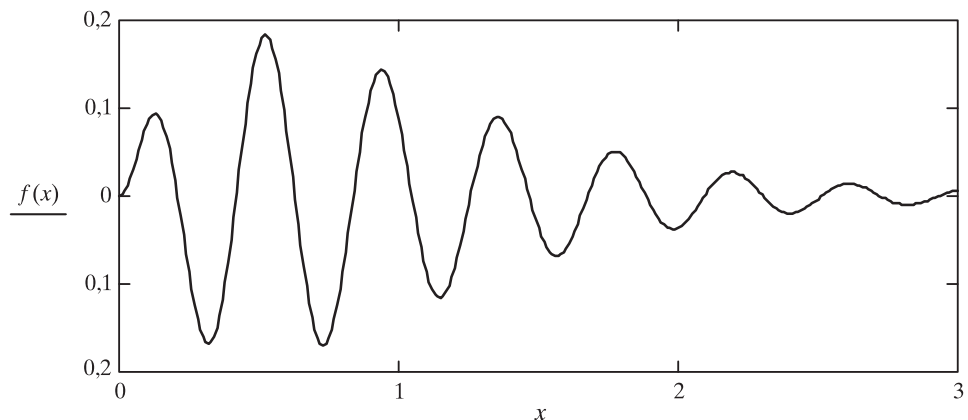
$$f(x) = xe^{-2x} \sin 15x.$$

Рассмотрим ее графики при различных изменениях x . Из первого графика видим, что глобальный экстремум находится в районах $x = -2$ и равен примерно 75.

$$x: = -2, -1, 99..4$$



Если смотреть другую область изменения, то глобальный экстремум находится в районе $x = 0,5$. Рассмотрим область изменения.



Используем метод Монте-Карло для нахождения глобального минимума функции. Сформируем два вектора X и Y , присвоив их нулевым элементам значение ноль

$$X_0 = 0; Y_0 = 0.$$

Зададим количеством случайных чисел n , которое мы будем использовать для вычисления минимума. Чем больше это количество, тем точнее будет результат вычисления: $n = 100000$.

С помощью функции $rnd(x)$ создадим вектор случайных значений элементов x . Функция $rnd(x)$ генерирует равномерно распределенные случайные числа в интервале $0 \dots x$.

Из графика видно, что нам достаточен интервал $0 \dots 3$.

$$i = 1..n, X_i = rnd(t).$$

Теперь в векторе X помещено 100000 случайных чисел. Вычислим значения функции от них и поместим их в вектор Y .

$$Y_i = f(X_i).$$

Величину минимального элемента этого вектора найдем, используя функцию \min .

$$Y_0 = \min(Y); Y_0 = -0,169.$$

Величину минимального элемента вектора X найдем, используя небольшую программу, и вычислим по ней ответ:

$$\begin{aligned} X_0 &= \text{for } i \in 1..n, \\ &\left| \begin{array}{l} X_0 \leftarrow X_i \text{ if } Y_0 = Y_i, \\ X_0, \end{array} \right. \\ X_0 &= 0,73. \end{aligned}$$

Получили первое приближенное значение минимума функции цели. Для уточнения значения используем градиентный метод. Поиск минимума ведется по следующим формулам:

$$x_{n+1} = x_n - \alpha_n g(x_n),$$

α_n выбираем из условия $F(\alpha_n) = f(x_n - \alpha_n g(x_n))$, $\alpha_n \geq 0$, $F(\alpha_n) = \min(F(\alpha))$.

$K = 10^{-6}$ – параметр, определяющий погрешность поиска минимума.

$\phi = \frac{1 + \sqrt{5}}{2}$ – отношение золотого сечения.

$fib(n) = \frac{\phi^{n+1} - (-\phi)^{-n-1}}{\sqrt{5}}$ – формула Бине для вычисления чисел Фибоначчи, где n – номер числа.

Сделаем подпрограмму для вычисления частной производной функции f в точке, заданной вектором v по переменной n .

Подпрограмма выглядит следующим образом:

$$proizv(v, n, f) = \left| \begin{array}{l} h \leftarrow 10^{-10} \\ v_1 \leftarrow v \\ v_{1_n} \leftarrow v_n + h \\ p \leftarrow \frac{f(v_1) - f(v)}{h} \\ p \end{array} \right.$$

Сделаем подпрограмму для вычисления значений проекций градиента на оси координат. Подпрограмма возвращает вектор значений проекций и использует подпрограмму вычисления частной производной:

$$grad(v, f) = \left| \begin{array}{l} n \leftarrow rows(v) \\ for\ i \in 0..n - 1 \\ g_i \leftarrow proizv(v, i, f) \\ g \end{array} \right.$$

Функция F , используемая для выбора α :

$$F(a, v, f) = \left| \begin{array}{l} n \leftarrow rows(v) \\ for\ i \in 0..n - 1 \\ x_i \leftarrow v_i - a \times grad(v, f) \\ f(x) \end{array} \right.$$

Далее сделаем подпрограмму поиска минимума функции одной переменной по методу Фибоначчи (одномерной оптимизации функции цели).

Подпрограмма поиска минимума с помощью метода градиентного спуска:

$$optND(v, f) = \left| \begin{array}{l} e \leftarrow 1 \\ while\ e > K \\ \alpha \leftarrow f\ opt(v, f) \\ for\ i \in 0..rows(v) - 1 \\ x_i \leftarrow v_i - \alpha \times grad(v, f)_i \\ e \leftarrow \sum_{i=0}^{rows(v)-1} |v_i - x_i| \\ v \leftarrow x \\ v \end{array} \right.$$

$$f \text{ opt}(v, f) = \left| \begin{array}{l} n \leftarrow 1000 \\ beg \leftarrow 0 \\ end \leftarrow 1 \\ x1 \leftarrow \frac{fib(n-2)}{fib(n)} \\ x2 \leftarrow \frac{fib(n-1)}{fib(n)} \\ \text{for } i \in 2..n-1 \\ \quad \left| \begin{array}{l} r \leftarrow \frac{fib(n-1-i)}{fib(n)} \\ end \leftarrow x2 \text{ if } F(x1, v, f) < \\ beg \leftarrow x1 \text{ if } F(x1, v, f) > \\ \text{if } F(x1, v, f) = F(x2, v, f) \\ \quad \left| \begin{array}{l} end \leftarrow x2 \\ beg \leftarrow x1 \\ x1 \leftarrow beg + r \\ x2 \leftarrow end - r \\ \text{(break) if } \|x1\| - \|x2\| \\ \frac{x1 + x2}{2} \end{array} \right. \end{array} \right. \end{array} \right.$$

Сформулируем еще раз нашу многоэкстремальную функцию:

$$f(x) = x_0 e^{-2x_0} \sin 15x_0, \\ x_0 = X_0.$$

Находим, в какой точке x достигается минимум функции цели по нашей программе:

$$optND(x, f) = (0, 73).$$

Находим значение минимума функции цели:

$$f(optND(x, f)) = -0,169.$$

Отсюда можно сказать, что метод градиентного спуска подтвердил сходимость решения методом Монте-Карло [2, 10].

Заключение. Целью нашего рассуждения и последующего исследования в работе является сведение полученной задачи НМП (18) к классической задаче математического программирования (2) для той же целевой функции f_0 , [2, 5, 6, 10].

Приложение

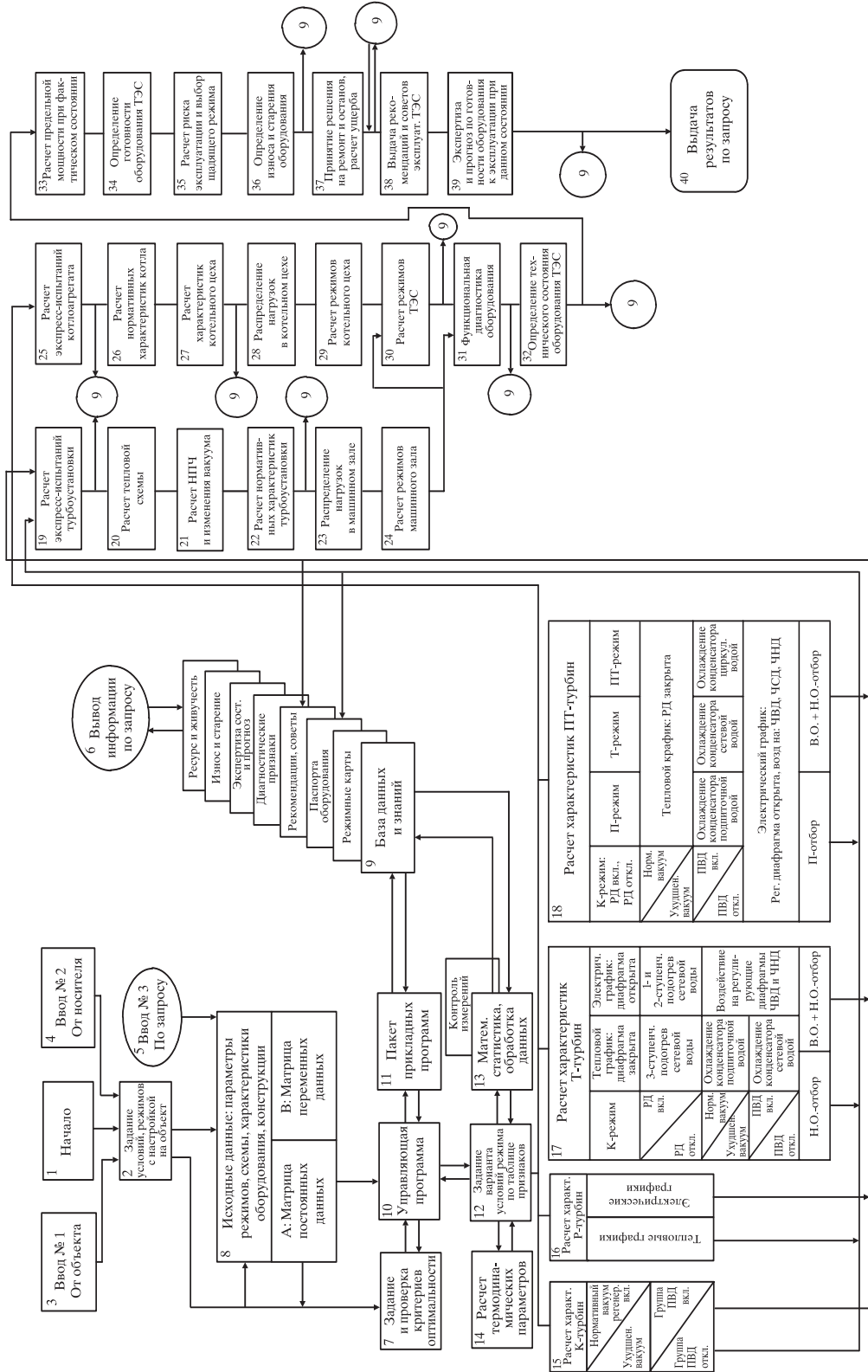


Рис. 1. Принципиальная модульно-структурная схема ВК для SKAIS ТЭС с теплофикационными установками

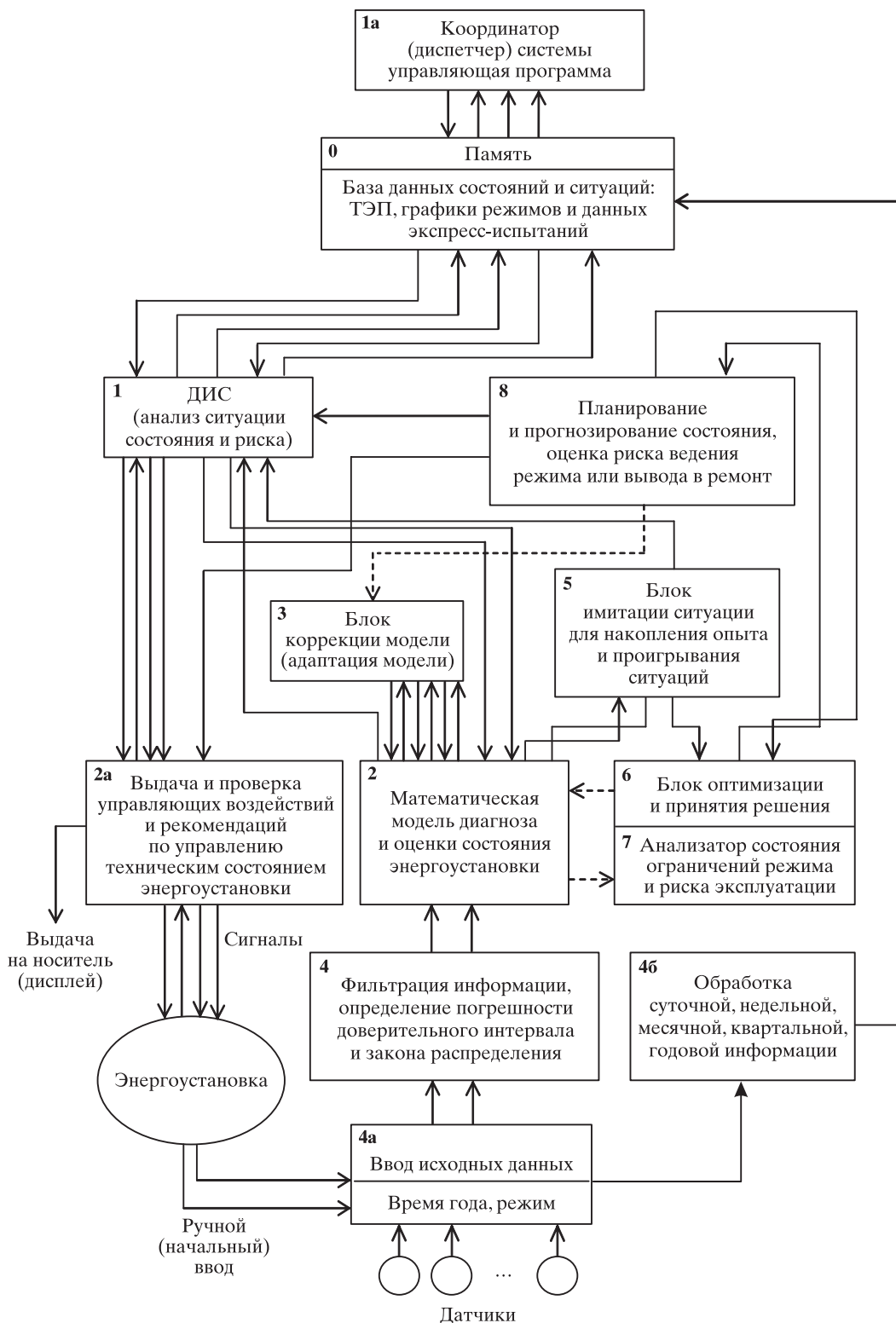


Рис. 2. SKAIS – подсистема диагностики состояния энергоустановки в контуре управления электростанции

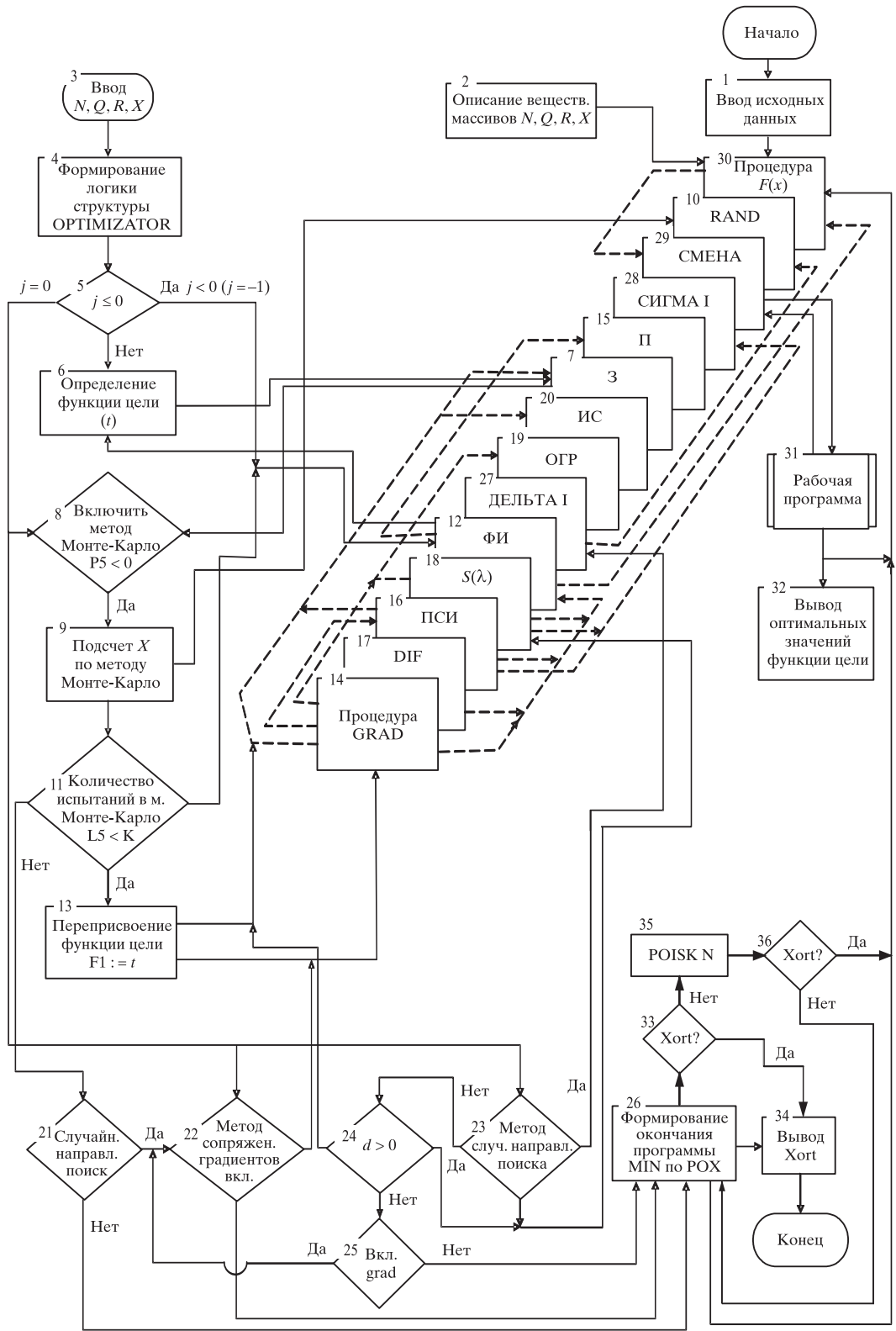


Рис. 3. Принципиальная блок-схема MIN + POISK N модуля OPTIMIZATOR

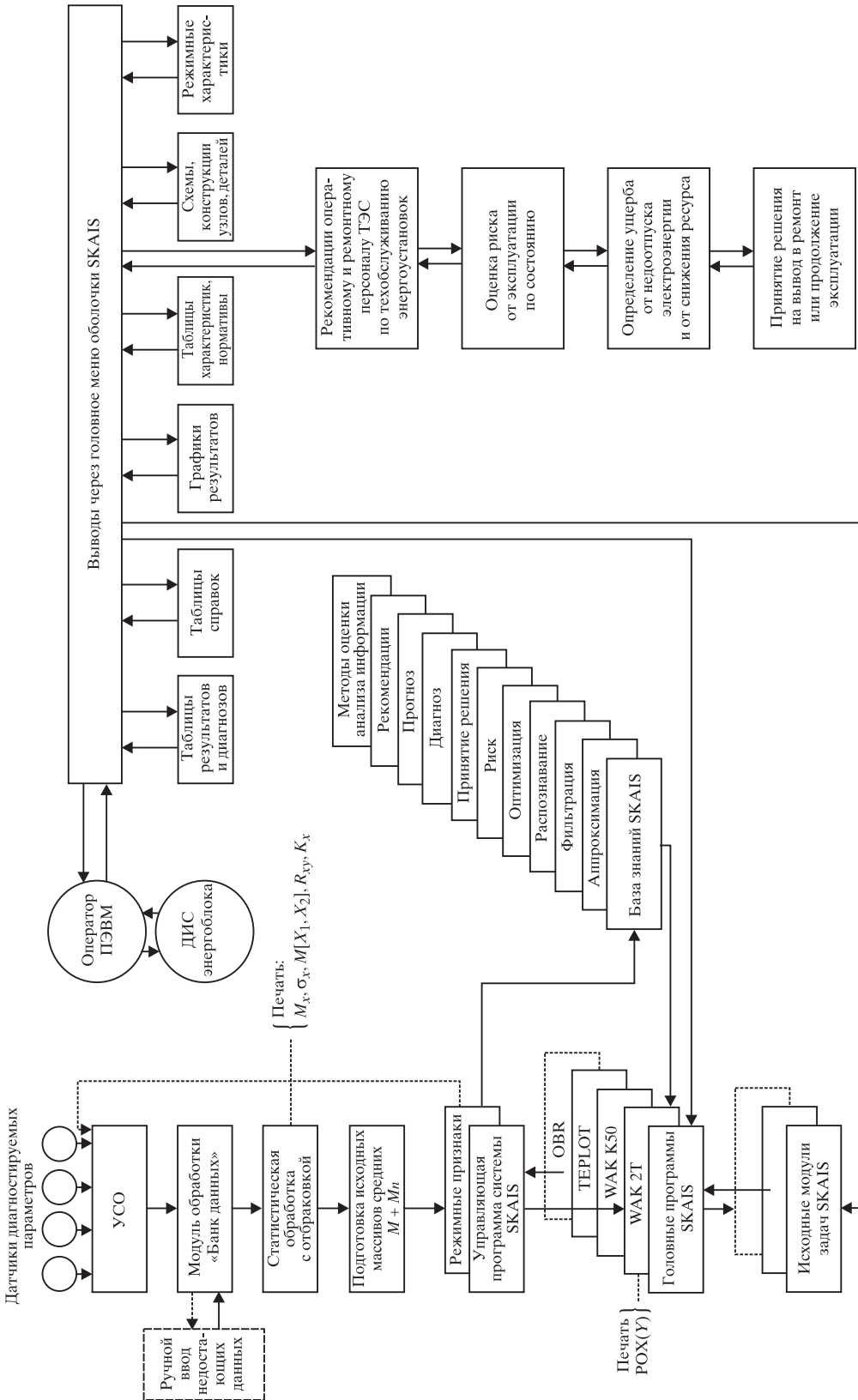


Рис. 4. Диагностический комплекс SKAIS (реализованный вариант)

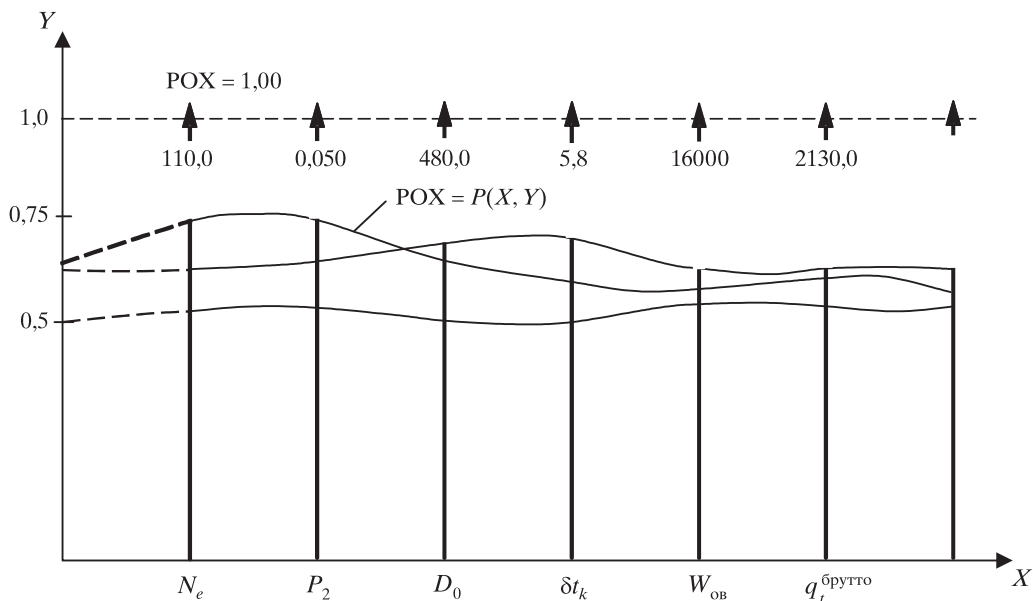


Рис. 5. «Похожесть» диагностируемого состояния (при сравнении с нормативным значением $POX = 1,00$) и определение фактического значения POX (общее состояние энергоустановки) как расстояния между ними, определяемое по формуле [4, 5, 9]:

$$POX = P(X, Y) = 1 - (2/N) \left[\sum_{i=1}^N (|X_i - Y_i| / (|X_i| + |Y_i|)) \right].$$

Здесь X – вектор измеренных параметров; Y – вектор эталонных значений параметров; N – количество анализируемых параметров; X_i, Y_i – наборы значений признаков (параметров состояния) для диагностируемого (X_i) и эталонного (Y_i) объектов (агрегатов). При этом, $i = \overline{1, N}$, $P(X, X) = 1$ и $P(X, Y) = P(Y, X)$. Если известно m элементов – эталонов Y^j , где $j = \overline{1, m}$, то, используя понятие «похожесть», можно найти ближайший к данному объекту X (его состоянию) эталон Y^k по максимуму значения коэффициента $P(X, Y^k)$, где $1 \leq k \leq m$.

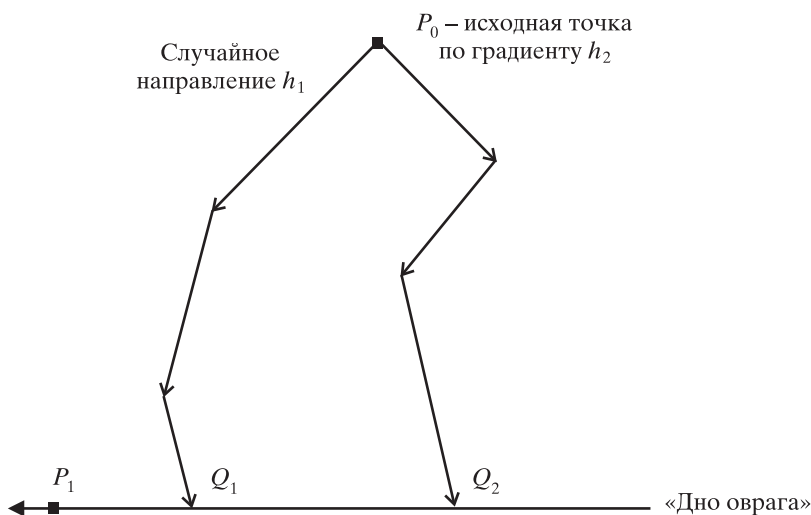


Рис. 6. Элемент многомерного поиска оптимума (по схеме MIN + POISK N)

Таблица П.1

Массив «весов» параметров-признаков (отклонение мощности турбины ΔN_3 и ее экономичности Δq_T^{dp} от гарантийного значения, DELTA = $\Delta N_3 / \Delta q_T^{dp}$) турбоустановки Т – 100–130 ТМЗ ст. № 7 Н ТЭЦ-4 (в отдельных опытах до и после ремонта) и сравнение параметров состояний по мере «похожести» (РОХ). Режим работы – конденсационный

Номинальные (гарантийные) параметры		$N_3^{ном}$	$P_0^{ном}$	$t_0^{ном}$	$P_2^{ном}$	$D_0^{ном}$	$\delta t_k^{ном}$	$W_0^{ном}$	$q_T^{бр(ном)}$	РОХ	DELTA
		МВт	МПа	°С	МПа	т/ч	°С	м³/ч	кДж/кВт·ч	–	МВт/кДж/кВт·ч
		110,0	12,8	555,0	0,0049	480,0	5,8	16000	8918	1,00	–
Фактические параметры	перед ремонтом	79,73	12,29	552,4	0,0218	392,6	22,5	13430	9245	0,52	
		75,09	12,29	549,8	0,0216	366,6	14,2	11762	9312	0,65	
		64,5	12,87	546,0	0,019	326,6	12,6	10151	9513	0,61	
		55,84	12,94	547,7	0,0156	302,2	13,4	9608	9689	0,57	
	после ремонта	83,1	12,8	550	0,0121	330,8	24,7	15300	8818	0,66	0,52/10,05
		85,7	12,9	543	0,0059	339,7	22,2	15900	8843	0,64	0,61/15,9
		78,0	12,8	556	0,0052	323,6	22,2	14420	8819	0,67	0,39/14,2
		71,1	13,04	551	0,0056	289,0	24,3	15608	8887	0,61	0,92/10,9
	82,3	12,92	543	0,0061	332,7	17,1	15700	8829	0,66	0,63/15,9	

Здесь под РОХ («похожесть») понимается расстояние между признаками (точнее их совокупностью) образов состояний энергоустановки близких номинальному (нормативному) состоянию. Похожесть фактического состояния энергоустановки номинальному состоянию определяется по формуле:

$$POX = P(X, Y) = 1 - (2/N) \left[\sum_{i=1}^N (|X_i - Y_i| / (|X_i| + |Y_i|)) \right],$$

где X_i, Y_i – соответственно наборы (векторы) значений признаков (параметров сравниваемых состояний) для диагностируемого и эталонного объектов (агрегатов). Здесь $i = \bar{1}, N, P(X, X) = 1$ и $P(X, Y) = P(Y, X)$. Если известно m элементов-эталонов Y^j , где $j = \bar{1}, m$, то, используя понятие «похожесть», можно найти ближайший к данному объекту X (его состоянию) эталон Y^k по максимуму значения коэффициента $P(X, Y^k)$, где $1 \leq k \leq m$.

Литература

1. Беллман Р, Заде Л. Принятие решений в расплывчатых условиях // Вопросы анализа и процедуры принятия решений. М.: Мир, 1976. С. 172–215.
2. Гилл Ф., Мюррей У, Райт М. Практическая оптимизация. М.: Мир, 1985. 509 с.
3. Зайченко Ю.П. Исследование операций: нечеткая оптимизация. К.: Выща шк., 1991. 191 с.
4. Крохин Г.Д., Супруненко М.Я., Манусов В.З. Распознавание образов при диагнозе состояния энергоустановок электростанций / Тр. 3-й Междунар. науч.-техн. конф. «АПЭП-96». Новосибирск: НГТУ, 1996.
5. Крохин Г.Д., Мухин В.С., Судник Ю.А. Интеллектуальные технологии в теплоэнергетике. Ч. 1. М.: ООО «УМЦ «Триада», 2010. 170 с.
6. Максимей И.В. Имитационное моделирование на ЭВМ. М.: Радио и связь, 1988. 232 с.
7. Растрингин Л.А. Статистические методы поиска. М.: Наука ФМЛ, 1968. 376 с.
8. Растрингин Л.А. Системы экстремального управления. М.: Наука ФМЛ, 1974. 632 с.
9. Супруненко М.Я. Эвристический поиск при разработке алгоритмов распознавания образов и оптимизации. Новосибирск: Сова, 2004. 72 с.

10. *Taha X.* Введение в исследование операций (в 2-х кн.). М.: Мир, 1985. Кн. 1. 479 с.; Кн. 2. 496 с.
11. *Штойер Р.* Многокритериальная оптимизация. Теория, вычисления и приложения. М.: Радио и связь, 1992. 504 с.

Bibliography

1. *Bellman R., Zade L.* Prinjatje reshenij v rasplyvchatyh uslovijah // Voprosy analiza i procedury prinjatija reshenij. М.: Mir, 1976. P. 172–215.
2. *Gill F, Mjurrej U., Rajt M.* Prakticheskaja optimizacija. М.: Mir, 1985. 509 p.
3. *Zajchenko Ju.P.* Issledovanie operacij: nechetkaja optimizacija. К.: Vywa shk., 1991. 191 p.
4. *Krohin G.D., Suprunenko M.Ja., Manusov V.Z.* Raspoznavanie obrazov pri diagnoze sostojanija jenergoustanovok jelektrostancij / Tr. 3-j Mezhdunar. nauch.-tehnič. konf. «APJeP-96». Novosibirsk: NGTU, 1996.
5. *Krohin G.D., Muhin V.S., Sudnik Ju.A.* Intellektual'nye tehnologii v teplojenergetike. Ch. 1. М.: ООО «UMC «Triada», 2010. 170 p.
6. *Maksimej I.V.* Imitacionnoe modelirovanie na JeVM. М.: Radio i svjaz', 1988. 232 p.
7. *Rastrigin L.A.* Statisticheskie metody poiska. М.: Nauka FML, 1968. 376 p.
8. *Rastrigin L.A.* Sistemy jekstremal'nogo upravlenija. М.: Nauka FML, 1974. 632 p.
9. *Suprunenko M.Ja.* Jevristicheskij poisk pri razrabotke algoritmov raspoznavanija obrazov i optimizacii. Novosibirsk: Sova, 2004. 72 p.
10. *Taha H.* Vvedenie v issledovanie operacij (v 2-h kn.). М.: Mir, 1985. Кн. 1. 479 p.; Кн. 2. 496 p.
11. *Shtojer R.* Mnogokriterial'naja optimizacija. Teorija, vychislenija i prilozhenija. М.: Radio i svjaz', 1992. 504 p.