

УДК 004.023 + 517.977.5 + 621.311.1

ИССЛЕДОВАНИЕ ДИНАМИЧЕСКИХ СВОЙСТВ ПОПУЛЯЦИОННЫХ АЛГОРИТМОВ В ЗАДАЧЕ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ИСТОЧНИКАМИ РЕАКТИВНОЙ МОЩНОСТИ В СИСТЕМАХ ЭЛЕКТРОСНАБЖЕНИЯ**В.З. Манусов, П.В. Матренин***Новосибирский государственный технический университет*

В работе рассматривается задача оперативного управления источниками компенсации реактивной мощности для сохранения устойчивости сети и минимизации потерь активной мощности. Эффективность компенсации зависит от расположения компенсирующих установок в узлах сети. Во многих работах высокую эффективность в решении подобных оптимизационных задач показывают популяционные алгоритмы, такие как генетический алгоритм, роевые алгоритмы. Новизна данной работы заключается в исследовании влияния динамических изменений условий задачи на эффективность указанных алгоритмов и определении наилучших стратегий учета подобных изменений. Для динамических изменений условий задач было использовано моделирование отказов компенсирующих установок. Оперативное управление мощностями компенсирующих установок выполнялось с помощью генетического алгоритма, алгоритма роя частиц и алгоритма роя пчел для сравнения их эффективностей и исследования влияния изменений условий задачи на работу алгоритмов. При решении динамических задач оптимизации рассматриваемые алгоритмы могут работать либо постоянно, реагируя на изменения условий, либо же перезапускаться в случае изменения. Проведенные в работе эксперименты позволили определить, что для алгоритма роя частиц необходимо выполнять перезапуск для выхода из локального экстремума. Для алгоритма роя пчел и генетического алгоритма перезапуск не требуется. При этом алгоритм роя пчел показал наилучшие результаты для рассматриваемой модели.

Ключевые слова: системы электроснабжения, глубокая компенсация, роевой интеллект, динамическая оптимизация, оперативное управление.

DOI: 10.17212/1727-2769-2017-3-74-87

Введение

Данная работа посвящена задаче уменьшения потерь энергии в системах электроснабжения до 10 кВ с помощью использования компенсирующих установок (КУ) для глубокой компенсации реактивной мощности. В настоящее время к проблеме снижения потерь электроэнергии подходят с разных точек зрения:

- внедрение современных энергосберегающих технологических установок и энергосберегающей электротехнической продукции;
- разработка интеллектуальных сетей и комплекса электрооборудования (воздушных линий передачи, трансформаторов, выключателей и т. д.) с использованием новых принципов, технологий передачи и управления процессом;
- снижение потребления реактивной энергии от генераторов электрических станций установкой источников реактивной мощности у потребителей (глубокая компенсация).

Повышение энергоэффективности также помогает сохранению окружающей среды. Например, расчеты по Германии показывают, что установка устройств компенсации реактивной мощности позволила в 2007 году снизить убытки примерно на 5,5 млрд киловатт часов и тем самым избежать выбросов почти 3 млн тонн CO₂ в год [1].

Доказано, что в радиальных сетях электроснабжения предприятий внедрение глубокой компенсации реактивной мощности в долгосрочной перспективе всегда выгодно и окупается за 1–4 года, поскольку затраты на КУ разовые, а экономия электроэнергии увеличивается пропорционально времени работы системы [2]. Поэтому в данной работе рассматривается не задача размещения КУ в узлах сети, а задача оперативного регулирования мощностей КУ для обеспечения минимальных потерь активной мощности с сохранением устойчивости системы.

Для построения математической модели задачи оптимизации необходимо задать критерий оптимальности, множество альтернатив и ограничений. Целью оптимизации является минимизация потерь активной мощности. Управляемыми переменными, определяющими множество альтернатив, являются значения мощностей КУ в узлах сети. Кроме того, имеется ограничение на $\operatorname{tg}(\varphi)$ (коэффициента мощности как отношения реактивной мощности к активной), его значение должно быть не меньше 0,1, поскольку слишком низкое значение может привести к неустойчивости системы. Второе ограничение связано с величиной компенсируемой мощности в узле, так как мощность КУ не может превышать потребляемую узлом реактивную мощность.

В результате сформулирована следующая задача оптимизации:

$$\begin{aligned} \Delta P(Q_{RPCU}) &\rightarrow \min, \\ Q_{RPCU} &= \{Q_1, Q_2, \dots, Q_n\}, \\ 0 < Q_i < Q_{\max i}, \quad i &= 1, \dots, n, \\ \operatorname{tg}(\varphi) &> 0,1, \end{aligned} \quad (1)$$

где $\Delta P(Q_{RPCU})$ – общие потери активной мощности в сети при данных мощностях КУ; Q_{RPCU} – вектор мощностей КУ; Q_i – мощность КУ в i -м узле (если $Q_{\max i} = 0$, то в i -м узле не устанавливается КУ); n – количество узлов, в которых установлены КУ, для рассматриваемой секции таких узлов 14.

В работе рассматривается подсистема электроснабжения одного предприятия по обогащению урана и изготовлению гексафторида урана в городе Ангарске. Система электроснабжения представляет собой подстанцию 0,4 кВ из четырех секций, каждая секция имеет радиальную схему. Для распределения электроэнергии по потребителям применены силовые распределительные пункты, которые запитаны от сборных шин секции подстанции. В каждой секции рассмотрено по десять наиболее загруженных шкафов распределительных, от которых осуществляется электроснабжение асинхронных электродвигателей системы вентиляции, электродвигателей насосов мощностью порядка 150 кВт и магистралей питания подкачивающих компрессоров. Каждая магистраль снабжает от 10 до 15 компрессоров с асинхронными электродвигателями мощностью 9 кВт, которые запитаны шлейфом. Потери активной мощности в линиях передач высоки из-за большой разветвленности сети и больших расстояний между узлами, связанной с технологическими особенностями рассматриваемого предприятия. Общая протяженность линий секции примерно 5 км. Полная система состоит из 8 подобных секций.

Предполагается глубокая компенсация реактивной мощности, следовательно, установка источников реактивной мощности – рядом со шкафами распределительными, для насосов – рядом со шкафами управления электродвигателями, для магистралей компрессоров – рядом со стойкой управления первого компрессора в магистрали.

В сети могут динамически меняться нагрузки и происходить отказы оборудования, поэтому необходимо выполнять регулирование мощностей КУ в реальном времени. Эффективность компенсации зависит от расположения компенсирующих установок в узлах сети. Как правило, в исследованиях, посвященных оптимальному размещению компенсирующих установок, эта задача рассматривается как статическая детерминированная задача оптимизации, которая может быть решена эвристическими методами. Но в технических системах при решении подобных задач целесообразно принимать во внимание их динамические и стохастические свойства, которые возникают в результате отказов оборудования и изменений режимов работы.

Для задач оптимизации с динамически меняющимися условиями необходимо применять алгоритмы, способные, во-первых, к самоорганизации, т. е. к автоматической адаптации к условиям задач; во-вторых, дающие эффективные решения достаточно быстро для работы в реальном времени. Наиболее известными алгоритмами, обладающими такими свойствами, являются генетический, поиск с запретами, имитация отжига, рой частиц и муравьиный алгоритм. Для задач оптимального размещения КУ в сетях электроснабжения успешно применяют генетический алгоритм [3, 4, 5], алгоритмы роевого интеллекта [2] и поиск с запретами [5].

В данной работе проводится сравнение генетического алгоритма, алгоритма роя частиц и роя пчел в случае непредсказуемо меняющихся во времени условий задачи. На данном этапе исследований выбран наиболее простой для проведения экспериментов вариант, связанный с отказами КУ.

1. Метод решения

1.1. Алгоритмы роевого интеллекта

Методы стохастической оптимизации являются особенно продуктивными в задачах, которые имеют такие особенности как нелинейность, недифференцируемость, недетерминированность, динамические свойства и высокую вычислительную сложность. Методы, использующие природные принципы, такие как эволюционные вычисления, роевой интеллект и метод имитации отжига, показывают наилучшее качество среди прочих стохастических методов. Эволюционные и роевые методы относятся к классу так называемых популяционных методов, поскольку используют системы агентов (популяции). Термин «агент» означает некоторую позицию в пространстве поиска решений, т. е. некоторое решение оптимизационной задачи. Процесс оптимизации может быть эволюционным или роевым:

- эволюционный процесс основан на создании новых популяций на каждом шаге алгоритма с учетом опыта, полученного предыдущими популяциями (естественный отбор в природе);
- роевой процесс означает перемещения агентов в пространстве поиска решений с использованием ряда правил и косвенного обмена данными между агентами. В отличие от эволюционных алгоритмов, агенты не создаются и не уничтожаются, а популяция не имеет никакой централизованной системы управления (колония муравьев, стая птиц).

Главное свойство популяционных алгоритмов – самоорганизация. Это приводит к способности алгоритмов исследовать пространство поиска решений независимо его размерности и топологии, что, в свою очередь, обеспечивает гибкость и способность относительно быстро находить близкие к глобально-оптимальным решения. Ниже даны краткие описания алгоритмов роя частиц, роя

пчел и генетического алгоритма. Подробные описания природных основ этих алгоритмов опущены, их можно найти в [6, 7]. Для более удобного применения положено, что в алгоритмах пространство поиска решений ограничено от 0 до 1 по каждой оси, а при вычислении целевой функции задачи (1) каждая координата x_i умножается на соответствующий коэффициент.

1.2. Алгоритмы роя частиц

Алгоритм роя частиц впервые был предложен в 1995 году в работах Дж. Кеннеди и Р. Эберхарта [8]. Алгоритм основан на принципах, по которым действует стая птиц. Птичья стая действует скоординированно, и каждая птица действует по простым правилам, наблюдая за другими птицами и согласовывая свои перемещения с ними. Согласно схеме описания роевых алгоритмов [6], алгоритм роя частиц PSO можно представить кортежем $\{S, M, A, P, I, O\}$.

1. Множество агентов (частиц) $S = \{s_1, s_2, \dots, s_{|S|}\}$. На j -й итерации i -я частица характеризуется состоянием $s_{ij} = \{X_{ij}, V_{ij}, X_{ij}^{best}\}$, где $X_{ij} = \{x_{ij}^1, x_{ij}^2, \dots, x_{ij}^l\}$ – вектор варьируемых параметров (положение частицы), $V_{ij} = \{v_{ij}^1, v_{ij}^2, \dots, v_{ij}^l\}$ – вектор скоростей частицы, $X_{ij}^{best} = \{b_{ij}^1, b_{ij}^2, \dots, b_{ij}^l\}$ – наилучшее по значению фитнес-функции положение частицы среди всех положений, которые она занимала в процессе работы алгоритма от 1-й до j -й итерации, l – количество управляемых переменных.

2. Вектор $M = X_j^{best}$ – наилучшее значение вектора среди всех частиц.

3. Алгоритм A описывает механизмы функционирования роя частиц.

3.1. Генерация начальных положений и скоростей ($j = 1$):

$$\begin{aligned} X_{i1} &= \text{rand}(0,1), i = 1, \dots, |S|, \\ V_{i1} &= \text{rand}(0, v_{\max}), i = 1, \dots, |S|, \\ X_{i1}^{best} &= X_{ij}, i = 1, \dots, |S|, \end{aligned}$$

где $\text{rand}(0,1)$ – вектор случайных чисел, равномерно распределенных от 0 до 1.

3.2. Вычисление фитнес-функций и определение наилучшего положения.

$$\begin{aligned} X_{ij}^{best} &= X_{ij} | f(X_{ij}^{best}) < f(X_{ij}), i = 1, \dots, |S|, \\ M &= X_{ij} | f(M) < f(X_{ij}), i = 1, \dots, |S|. \end{aligned}$$

3.3. Перемещения частиц (2):

$$\begin{aligned} V_{ij+1} &= V_{ij}\omega + \alpha_1(X_{ij}^{best} - X_{ij})\text{rand}(0,1) + \\ &+ \alpha_2(M - X_{ij})\text{rand}(0,1) \quad (i = 1, \dots, |S|); \end{aligned}$$

$$V_{ij+1} = V_{\max} | V_{ij+1} > v_{\max}, i = 1, \dots, |S|;$$

$$V_{ij+1} = -V_{\max} | V_{ij+1} < -v_{\max}, i = 1, \dots, |S|;$$

$$X_{ij+1} = X_{ij} + V_{ij+1}, i = 1, \dots, |S|;$$

$$X_{ij+1} = 1 \mid X_{ij+1} > 1, i = 1, \dots, |S|;$$

$$X_{ij+1} = 0 \mid X_{ij+1} < 0, i = 1, \dots, |S|.$$

Если не выполнено условие остановки, переход к п. 3.2.

4. Параметры $P = \{\alpha_1, \alpha_2, \omega, v_{\max}\}$. В данной статье значение каждого параметра не описывается, поскольку такое описание многократно дано в литературе, например [9, 10].

5. Идентификаторы I и O – вход и выход алгоритма роя частиц для взаимодействия с решаемой задачей.

1.3. Алгоритмы роя пчел

Алгоритм роя пчел разработан в 2005 году рядом авторов [11]. Он основан на симуляции поведения пчел при поиске нектара и косвенном обмене информацией между ними. Согласно схеме описания роевых алгоритмов [6], алгоритм роя пчел $ABCO$ можно представить кортежем $\{S, M, A, P, I, O\}$.

1. Множество агентов (пчел) $S = \{s_1, s_2, \dots, s_{|S|}\}$, при этом $s_{ij} = X_{ij} = \{x_{ij}^1, x_{ij}^2, \dots, x_{ij}^l\}$.

2. Средством косвенного обмена M является список лучших и перспективных позиций, найденных на j -й итерации, $M = \{N_{ij}^b, N_{kj}^g\}, i = 1, \dots, n^b, k = 1, \dots, n^g$.

3. Алгоритм A описывает механизмы функционирования роя пчел.

3.1. Инициализация начальных положений ($j = 1$) выполняется только для подмножества агентов, называемых разведчиками:

$$X_{i1} = \text{rand}(0, 1), i = 1, \dots, n^s,$$

где n^s – количество пчел-разведчиков.

3.2. Вычисление фитнес-функций $f(X)$ выполняется на первой итерации только для разведчиков, а на следующих – для всех агентов.

3.3. Миграция пчел. Среди всех агентов выбираются n^b агентов с наилучшими значениями целевой функции, а из оставшегося множества еще n^g агентов по тому же принципу. Используя эти два набора, формируются списки лучших и перспективных позиций $M = (N_{ij}^b, N_{kj}^g)$. При этом любыми двумя позициями в M должно быть не меньше rx . В окрестность каждой лучшей позиции отправляется c^b пчел, в окрестность каждой перспективной – c^g пчел:

$$X_{(i-1)cb+kj} = N_{ij-1}^b + \text{rand}(-1, 1) \text{rad}, i = 1, \dots, n^b, k = 1, \dots, c^b,$$

$$X_{nbc b+(i-1)cg+kj} = N_{ij-1}^g + \text{rand}(-1, 1) \text{rad}, i = 1, \dots, n^g, k = 1, \dots, c^g.$$

В конце итерации пчелы-разведчики отправляются в случайные позиции:

$$X_{nbc b+ngcg+ij} = \text{rand}(-1, 1) \text{rad}, i = 1, \dots, n^s.$$

Если не выполнено условие остановки, переход к п. 3.2.

4. Параметры $P = \{n^s, n^b, n^g, c^b, c^g, \text{rad}, rx\}$.

5. Идентификаторы I и O – вход и выход алгоритма, как и в PSO.

1.4. Генетический алгоритм

Генетический алгоритм начал широко применяться для решения оптимизационных задач в 1960–70-х гг. в результате работ Инго Рехенберга и Джона Голланда [12]. Алгоритм основан на моделировании механизмов эволюции, таких как наследственность, мутации, отбор и скрещивание. Агенты называются особями или хромосомами. В генетическом алгоритме нет средств косвенного обмена данными между агентами, а популяция агентов S создается вновь на каждой итерации. Так как генетический алгоритм намного более широко известен, чем алгоритмы роевого интеллекта, здесь давать его описание нецелесообразно, поэтому оно пропущено. Необходимо лишь указать, что используется классический алгоритм с одноточечным кроссинговером двух родителей.

1.5. Подбор параметров

Особенностью стохастических алгоритмов роя пчел и роя частиц, как и других алгоритмов роевого интеллекта, является необходимость настройки их эвристических параметров P (коэффициентов) под решаемый класс задач для получения более качественных решений. Это подтверждается рядом исследований [9, 10, 13, 14] и No Free Lunch теоремой [15]. Иными словами, алгоритмы роевого интеллекта не всегда работают эффективно и требуют настройки параметров для каждого вида задач. Наиболее эффективным методом настройки параметров является техника мета-оптимизации [13]. Мета-оптимизация предполагает рассмотрение задачи подбора параметров как отдельной задачи оптимизации. При этом некий алгоритм оптимизации решает прикладную задачу, а мета-оптимизация подбирает параметры этого алгоритма для получения наилучших решений прикладной задачи. В работе [14] для подбора параметров была использована техника эволюционной адаптации значений параметров под условия решаемых задач. Этот метод адаптации устойчив к перенастройке (overtuning), поэтому позволяет получать хорошие результаты не только на тех задачах, на которых выполнялся подбор параметров, но и на других подобных задачах. В данной работе были использованы параметры, найденные ранее в работе [2] для подобной задачи оптимизации.

1.6. Применение алгоритмов для рассматриваемой задачи

Для взаимодействия алгоритмов оптимизации с моделью рассматриваемой задачи необходимо задать отношение между критерием оптимальности в модели $W(Q_{RPCU})$ и в алгоритме $f(X)$. Для этого необходимо установить соответствие между позицией агента X и вектором, задающим размещение КУ в сети Q_{RPCU} . В данной работе вектор X использован не как вектор мощностей КУ, а как вектор коэффициентов, так что мощность каждой КУ определялась как произведение элемента вектора X на определенную в ходе расчетов максимально допустимую мощность КУ в соответствующем узле:

$$Q_i = x_i Q_{i \max}.$$

Если для рассматриваемого варианта размещения КУ $\text{tg}(\varphi)$ не подходит под ограничения, то к значению критерия оптимизации прибавляется штрафное значение, величина которого значительно больше возможных значений критерия.

2. Эксперименты

2.1. Цель экспериментов

Проведенный эксперимент моделировал ситуации отказа одной из КУ в сети. При этом необходимо в реальном времени провести автоматическую регулировку мощностей оставшихся КУ, для сохранения устойчивости системы и минимизации потерь активной мощности. В случае использования популяционных алгоритмов для решения динамически меняющихся задач существует два альтернативных пути.

1. Как только произошло изменение условий задачи, прервать процесс работы алгоритма и запустить его заново для новой задачи оптимизации. В этом случае динамически меняющаяся задача рассматривается как набор отдельных не связанных между собой задач статической оптимизации. Такой подход не учитывает опыт популяционного алгоритма, полученный до изменения условий задачи. Но зато позволяет алгоритму выйти из локального экстремума и начать процесс поиска с нуля, т. е. с инициализации случайной популяции. Будем называть этот путь «с рестартом».

2. Никогда не останавливать процесс работы алгоритма. Таким образом, при изменении условий задачи процесс поиска решения начнется не с нуля, а с того состояния, в котором популяция находилась в момент смены условий. При этом есть шанс быстро найти эффективное решение задачи с измененными условиями, но выше риск не выйти из текущего локального экстремума. Этот путь назовем «без рестарта».

Целью экспериментов является сравнение эффективности этих путей для рассматриваемых алгоритмов оптимизации. Априори было предположено, что для алгоритма роя частиц и генетического алгоритма лучше первый путь, а для алгоритма роя пчел – второй, поскольку только алгоритм роя пчел гарантирует, что все агенты популяции не окажутся в окрестности одного решения.

2.2. Описание эксперимента

Во время экспериментов задача оптимизации (1) решалась отдельно каждым из рассматриваемых алгоритмов. Параметры алгоритмов показаны в табл. 1.

Таблица 1 / Table 1

Параметры алгоритмов
Algorithms' parameters

Алгоритм	Количество агентов	Эвристические параметры
PSO	100	$\alpha_1 = 2,03$ $\alpha_2 = 2,32$, $\omega = 0,87$, $v_{\max} = 0,9$
ABCO	100	$n^s = 10$, $n^b = 15$, $n^g = 10$, $c^b = 4$, $c^g = 3$, $rad = 0.01$, $rx = 0,05$
GA	100	$p_{\text{crossover}} = 0,9$, $p_{\text{mutation}} = 0,2$

Эксперименты для каждого из алгоритмов выполнялись по следующей схеме.

1. Найти квазиоптимальное решение задачи (1) для случая, когда все КУ работоспособны.

2. Для каждого узла, в котором размещена КУ.

2.1. Положить, что в текущем узле КУ отказала и полностью перестала функционировать.

2.2. Продолжить процесс работы алгоритма оптимизации для решения задачи с измененными условиями, т. е. выбрать способ без рестарта, описанный в 3.1.

Процесс работы останавливается через 200 тысяч итераций после изменения условий задачи.

2.3. Запустить процесс работы алгоритма оптимизации с нуля без учета полученного ранее решения, т. е. использовать способ с рестартом, описанный в 3.1. Процесс работы останавливается через 200 тысяч итераций.

Так как решение задачи оптимизации предполагается в реальном времени, то важно, как быстро будет найдено квазиоптимальное решение задачи после изменения ее условий. Поэтому фиксировались результаты, полученные на 100, 500, 1000, 2000 итерациях. Окончательный результат, полученный после 200 000 итераций, представляет скорее теоретический интерес.

2.3. Результаты экспериментов

Результаты экспериментов показаны в табл. 2–4. Табл. 2 показывает результаты решения оптимизационной задачи (1) для случаев отказов разных КУ (приведены результаты не по всем КУ).

Таблица 2 / Table 2

Результаты решения задачи (1)

Solutions task (1)

Алгоритм	Способ	id отказавшей КУ	$\Delta P(100)$, кВ	$\Delta P(500)$, кВ	$\Delta P(1000)$, кВ	$\Delta P(2000)$, кВ	$\Delta P(20000)$, кВ
PSO	с рест.	1	312,02	311,92	311,92	311,92	311,90
PSO	без рест.	1	324,48	324,48	324,48	324,48	324,48
ABCO	с рест.	1	311,99	311,42	310,97	310,70	306,14
ABCO	без рест.	1	311,80	311,37	310,93	310,77	310,34
GA	с рест.	1	346,34	346,34	344,27	333,88	316,42
GA	без рест.	1	343,28	339,19	339,19	335,69	318,15
PSO	с рест.	7	311,32	311,31	311,31	311,31	311,31
PSO	без рест.	7	318,06	318,06	318,06	318,06	318,06
ABCO	с рест.	7	311,54	311,02	310,87	310,80	304,43
ABCO	без рест.	7	311,36	311,09	310,96	310,68	300,57
GA	с рест.	7	354,69	344,42	344,42	338,26	315,68
GA	без рест.	7	345,26	345,26	345,26	342,53	319,40
PSO	с рест.	9	352,67	352,67	352,67	352,67	352,67
PSO	без рест.	9	392,46	392,46	392,46	392,46	392,46
ABCO	с рест.	9	324,59	324,09	323,90	323,55	315,19
ABCO	без рест.	9	324,19	323,54	323,36	323,15	321,44
GA	с рест.	9	392,87	392,87	392,87	392,87	368,21
GA	без рест.	9	408,93	396,98	387,31	387,31	366,19
PSO	с рест.	12	358,78	358,78	358,78	358,78	358,78
PSO	без рест.	12	398,97	398,97	398,97	398,97	398,97
ABCO	с рест.	12	330,12	329,61	329,36	329,03	318,74
ABCO	без рест.	12	329,83	329,06	328,83	328,63	321,72
GA	с рест.	12	409,76	407,65	399,73	392,86	382,14
GA	без рест.	12	415,28	409,39	408,97	399,58	377,26
PSO	с рест.	13	311,57	311,57	311,56	311,56	311,56
PSO	без рест.	13	322,89	322,89	322,89	322,89	322,89
ABCO	с рест.	13	311,58	311,40	311,33	311,23	300,08
ABCO	без рест.	13	311,07	310,91	310,91	310,76	304,81
GA	с рест.	13	352,56	342,97	340,22	332,53	312,68
GA	без рест.	13	344,40	344,40	344,40	327,93	319,65

В первом столбце указан используемый алгоритм. Во втором – используемый способ учета меняющихся условий задачи, «с рест.» – с рестартом, «без рест.» – без рестарта (см. 2.1). В третьем столбце указан номер отказавшей в данном эксперименте КУ. Следующие столбцы показывают полученные значения критерия

оптимальности (потери активной мощности в кВт) после указанного в скобках количества итераций алгоритма. Отсчет итераций велся после изменения условия задачи, т. е. от шага 2.1 в описании эксперимента в п. 2.2.

Из табл. 2 видно, что почти во всех указанных случаях наилучшие решения дает алгоритм роя пчел независимо от числа итераций. Алгоритм роя частиц при использовании способа с рестартом дает результаты значительно лучшие, чем при использовании способа без рестарта. Причем результаты алгоритма роя частиц без рестарта практически не улучшаются с увеличением числа итераций, что говорит о неспособности алгоритма выйти из локального экстремума при изменении условий задачи.

Так происходит потому, что при сжатии роя частиц в окрестность некоторого экстремума элементы $(X^{best}_{ij} - X_{ij})$ и $(M - X_{ij})$ в формуле (2) оказываются очень малы и, следовательно, скорости частиц тоже оказываются слишком малы для выхода из этого экстремума за приемлемое время. Генетический алгоритм показал худший результат, а его эффективность для одних случаев лучше при использовании рестарта, а для других – без него. Генетический алгоритм способен выходить из локальных экстремумов с помощью мутации, в результате которой агенты популяции могут оказываться в абсолютно любых точках в пространстве решения.

В табл. 2 приведены подробные результаты только для некоторых КУ. Итоговые результаты, усредненные по всем КУ, показаны в табл. 3 и 4. В табл. 3 показаны отклонения критерия оптимальности (ΔP) от наилучшего значения, найденного среди всех алгоритмов после указанного числа итераций. Отклонения усреднены по всем КУ. В табл. 4 по той же схеме показаны максимальные отклонения.

Таблица 3 / Table 3

Сравнение эффективностей алгоритмов, средних отклонений
Comparison of the algorithms' efficiency, the average deviation

Алг.	Путь	Отклонение ΔP от наилучшего после заданного числа итераций, %				
		100	500	1000	2000	20000
PSO	с рест.	4,132	4,127	4,125	4,123	4,118
PSO	без рест.	8,025	8,024	8,024	8,024	8,024
ABCO	с рест.	2,728	2,652	2,612	2,563	0,597
ABCO	без рест.	2,617	2,551	2,522	2,475	0,492
GA	с рест.	17,057	14,606	13,486	11,323	6,530
GA	без рест.	13,952	13,064	12,633	11,687	6,897

Таблица 4 / Table 4

Сравнение эффективностей алгоритмов, максимальных отклонений
Comparison of the algorithms' efficiency, the maximum deviation

Алг.	Путь	Отклонение ΔP от наилучшего после заданного числа итераций, %				
		100	500	1000	2000	20000
PSO	с рест.	12,565	12,565	12,565	12,565	12,565
PSO	без рест.	25,173	25,173	25,173	25,173	25,173
ABCO	с рест.	4,072	3,982	3,9372	3,883	2,065
ABCO	без рест.	3,852	3,795	3,784	3,735	1,981
GA	с рест.	28,558	27,896	25,411	24,643	19,893
GA	без рест.	30,289	28,442	28,309	25,363	18,361

Эксперименты показали, что алгоритм роя частиц оказался наиболее чувствительным к выбору способа учета динамических изменений условий задач и следует

выполнять рестарт алгоритма, как только условия изменились. Для генетического алгоритма немного предпочтительнее выполнять рестарт, а для алгоритма роя пчел, наоборот, учет полученных ранее решений помогает при решении меняющихся задач. В целом для рассматриваемой задачи управления мощностями КУ алгоритм роя пчел показал существенно более высокую эффективность. Это можно объяснить тем, что агенты алгоритма роя пчел всегда рассредоточены по нескольким областям пространства поиска решений и никогда не собираются в окрестности одного экстремума. При этом алгоритм роя пчел несколько сложнее в реализации и существенно более трудоемкий с точки зрения подбора эвристических параметров, так как в алгоритме роя пчел 7 параметров, как показано в табл. 1.

Результаты, приведенные в табл. 3 и 4, показаны на рис. 1 и 2 соответственно. На приведенных гистограммах видно, что алгоритм роя частиц без рестарта показывает эффективность ниже, чем с рестартом, причем результаты не улучшаются с увеличением числа итераций. Это говорит о быстрой сходимости алгоритма к некоторому локальному экстремуму. Генетический же алгоритм улучшает свои результаты с увеличением числа итераций благодаря мутации, создающей новые варианты решений. Тем не менее его эффективность уступает алгоритмам роя частиц и роя пчел даже после 200 000 итераций. Алгоритм роя пчел быстро находит эффективное решение, но как видно из графиков, увеличение итераций с 2000 до 200 000 приводит к повышению эффективности. Это может быть объяснено влиянием пчел-разведчиков, выполняющих случайный поиск, эффективность которого линейно зависит от числа итераций и может оказать существенное влияние при большом числе итераций. Кроме того, случайный поиск защищает алгоритм роя пчел от преждевременной сходимости, которую показал алгоритм роя частиц.

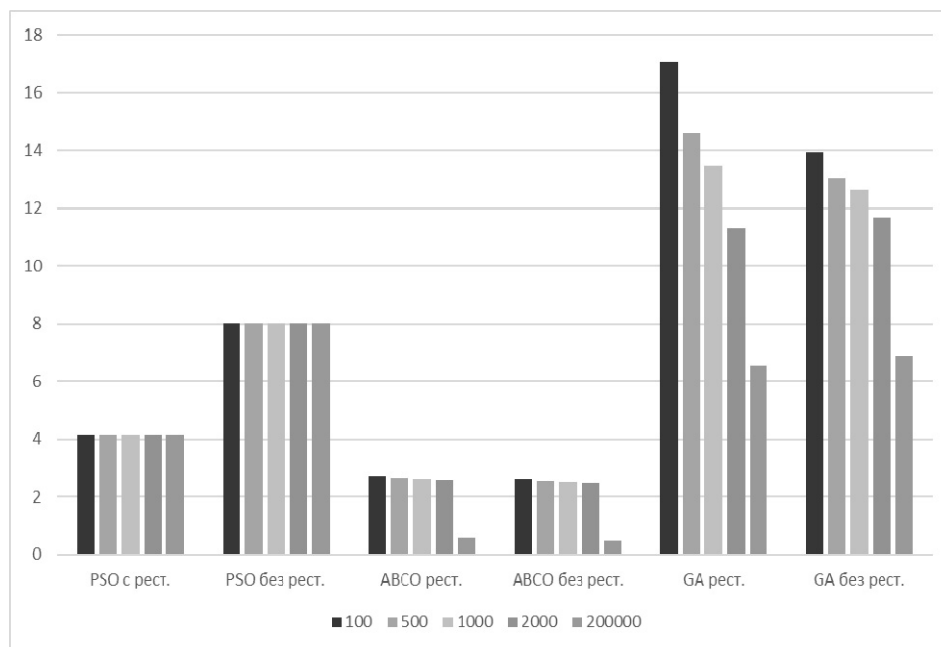


Рис. 1 – Сравнение эффективностей алгоритмов, средних отклонений

Fig. 1 – Comparison of the algorithms' efficiency, the average deviation

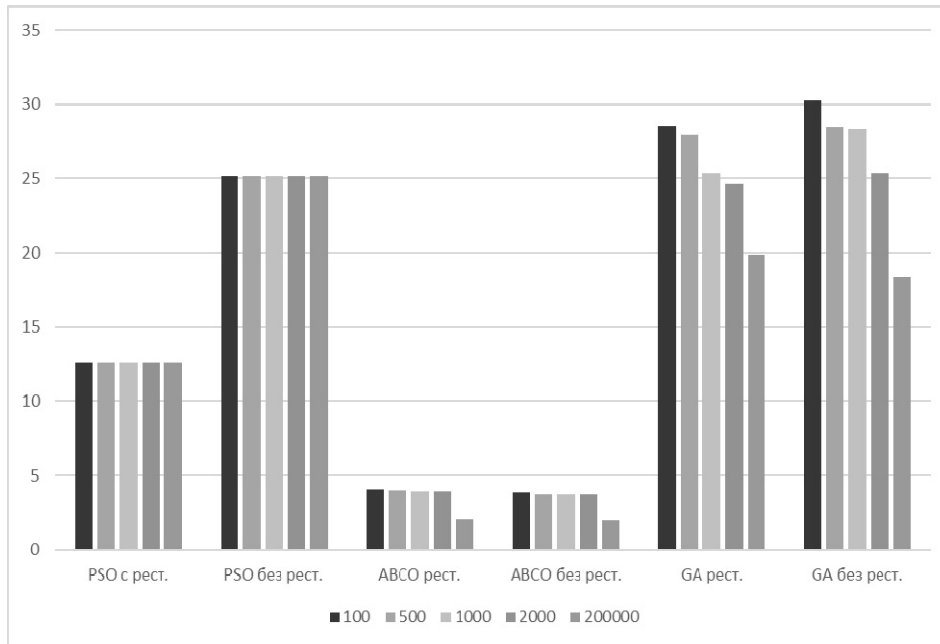


Рис. 2 – Сравнение эффективностей алгоритмов, максимальных отклонений

Fig. 2 – Comparison of the algorithms' efficiency, the maximum deviation

3. Заключение

1. Рассмотрена задача управления источниками реактивной мощности в сетях электроснабжения как динамическая задача оптимизации, решаемая в режиме реального времени. Критерием задачи является снижение потерь активной мощности в системе электроснабжения. В качестве управляемых переменных выбраны мощности компенсирующих установок. Для проведения экспериментов моделировались отказы КУ, приводящие к динамическим изменениям условий задачи.

2. Для решения данной задачи оптимизации были применены популяционные алгоритмы: генетический алгоритм, алгоритм роя частиц и роя пчел. При решении динамических задач оптимизации эти алгоритмы могут работать либо постоянно, реагируя на все изменения условий задачи, либо же перезапускаться при каждом изменении условий. Эти два варианта можно обозначить «без перезапуска» и «с перезапуском».

3. Эксперименты показали, что для алгоритма роя частиц необходимо выполнять перезапуск, иначе алгоритм не выходит из локального экстремума, так как агенты алгоритма роя частиц имеют тенденцию через определенное количество итераций собираться в окрестности одного экстремума (табл. 2, 3). Генетический алгоритм показывает одинаковую эффективность с перезапуском и без него, отклонение результатов не более 3,1 %. Результаты генетического алгоритма оказались на 2...10 % хуже, чем роевых алгоритмов (табл. 3, 4). Наилучшие результаты показал алгоритм роя пчел, при этом вариант без перезапуска оказался незначительно более эффективным (на 0,1 %). При правильно настроенных параметрах агенты алгоритма роя пчел всегда рассредоточены по нескольким экстремумам, что позволяет как учитывать ранее найденные решения, так и находить новые.

4. В настоящее время в работе принят ряд ограничений. Не используется модель отказов КУ, их распределения. Рассматриваются только последствия отказов. В дальнейшем планируется усложнить рассмотренную в статье задачу оптимизации и приблизить ее к реальным условиям. Для этого модель системы электрооборудования будет включать не только отказы КУ, которые на самом деле происходят редко, но и учитывать динамические изменения нагрузок в сетях и режимов работы потребителей.

ЛИТЕРАТУРА

1. Energieeinsparung durch Blindleistungskompensation [Electronic resource]. – URL: http://www.energie-im-unternehmen.de/downloads/energieeffizienz_steigerung_blindleistungskompensation.pdf (accessed: 28.11.2017).
2. **Manusov V., Tretyakova E., Matrenin P.** Population-based algorithms for optimization of the reactive power distribution and selection of the cable cross-section in the power-supply systems // *Applied Mechanics and Materials*. – 2015. – Vol. 792. – P. 230–236.
3. **Da Silva E.L., Gil H.A., Areiza J.M.** Transmission network expansion planning under an improved genetic algorithm // *IEEE Transactions on Power Systems*. – 2000. – Vol. 15, iss. 3. – P. 1168–1174.
4. Optimal location of phase shifters in the French network by genetic algorithm / P. Paterni, S. Vitet, M. Bena, A. Yokoyama // *IEEE Transactions on Power Systems*. – 1999. – Vol. 14, iss. 1. – P. 37–42.
5. **Mantawy A.H., Abdel-Magid Y.L., Selim S.Z.** Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem // *IEEE Transactions on Power Systems*. – 1999. – Vol. 14, iss. 3. – P. 829–836.
6. **Матренин П.В., Секаев В.Г.** Системное описание алгоритмов роевого интеллекта // *Программная инженерия*. – 2013. – № 12. – С. 39–45.
7. **Матренин П.В.** Описание и реализация алгоритмов роевого интеллекта с использованием системного подхода // *Программная инженерия*. – 2015. – № 3. – С. 27–34.
8. **Kennedy J., Eberhart R.** Particle swarm optimization // *Proceedings of IEEE International Conference on Neural Networks*. – 1995. – Vol. 4. – P. 1942–1948.
9. **Matrenin P.V., Sekaev V.G.** Particle swarm optimization with velocity restriction and evolutionary parameters selection for scheduling problem // *International Siberian Conference Control and Communications SIBCON 2015: proceedings, Russia, Omsk, 21–23 May 2015*. – Omsk, 2015. – P. 1–5.
10. **Манусов В.З., Матренин П.В.** Адаптивный алгоритм роя частиц в задачах оперативного планирования // *Вестник компьютерных и информационных технологий*. – 2016. – № 4 (142). – С. 11–15.
11. The Bees Algorithm – a novel tool for complex optimisation problems / D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi // *IPROMS 2006. Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems*. – Oxford, Elsevier, 2006. – P. 454–459. – doi: <https://doi.org/10.1016/B978-008045157-2/50081-X>.
12. **Holland J.H.** *Adaptation in natural and artificial systems*. – Ann Arbor: University of Michigan Press, 1975. – 225 p.
13. **Pedersen M., Chippereld A.** Simplifying particle swarm optimization // *Applied Soft Computing*. – 2010 – N 10 (2). – P. 618–628.
14. **Матренин П.В., Секаев В.Г.** Оптимизация адаптивного алгоритма муравьиной колонии на примере задачи календарного планирования // *Программная инженерия*. – 2013. – № 4. – С. 34–40.
15. **Wolpert D.H., Macready W.G.** No free lunch theorems for optimization // *IEEE Transactions on Evolutionary Computation*. – 1997. – Vol. 1, iss. 1. – P. 67–82.

RESEARCH ON DYNAMIC PROPERTIES OF POPULATION ALGORITHMS IN OPERATION CONTROL OF REACTIVE POWER UNITS IN SMART GRID

Manusov V.Z., Matrenin P.V.

Novosibirsk State Technical University, Novosibirsk, Russia

In this paper, the problem of operational control of reactive power compensation units is considered to maintain network stability and minimize the loss of active power. The efficiency of compensation depends on the location of the compensating installations in the network nodes. In many works, high-performance solutions to such optimization problems are shown by population-based algorithms, such as the Genetic algorithm and Swarm Intelligence algorithms. The novelty of this work is to study the effect of dynamic changes in the problem conditions on the effectiveness of these algorithms and to determine the best strategies for accounting for such differences. Compensation unit failures were simulated to change dynamically the problem conditions. Operational control of compensation unit powers was carried out using the Genetic algorithm, the Particle Swarm Optimization algorithm and the Artificial Bee Colony Optimization algorithm to compare their efficiencies and to study the effect of changes in the conditions of the problem on the algorithm performances. When solving dynamic optimization problems, the considered algorithms can work either permanently reacting to changes in conditions, or they can be restarted in the case of a change. Experiments allowed us to determine that for the Particle Swarm Optimization algorithm it is necessary to perform a restart for exiting the local extremum. However, restarting is not required for the Artificial Bee Colony Optimization algorithm and the Genetic algorithm. In this case, the Bee Swarm algorithm showed the best results for the model under consideration.

Keywords: dynamic optimization problems, operation control, smart-grid, swarm intelligence, stochastic optimization.

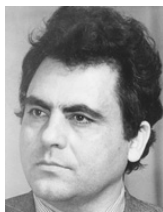
DOI: 10.17212/1727-2769-2017-3-74-87

REFERENCES

1. *Energieeinsparung durch Blindleistungskompensation*. Available at: http://www.energie-im-unternehmen.de/downloads/energieeffizienz_steigerung_blindleistungskompensation.pdf (accessed 28.11.2017).
2. Manusov V., Tretyakova E., Matrenin P. Population-based algorithms for optimization of the reactive power distribution and selection of the cable cross-section in the power-supply systems. *Applied Mechanics and Materials*, 2015, vol. 792, pp. 230–236.
3. Da Silva E.L., Gil H.A., Areiza J.M. Transmission network expansion planning under an improved genetic algorithm. *IEEE Transactions on Power Systems*, 2000, vol. 15, iss. 3, pp. 1168–1174.
4. Paterni P., Vitet S., Bena M., Yokoyama A. Optimal location of phase shifters in the french network by genetic algorithm. *IEEE Transactions on Power Systems*, 1999, vol. 14, iss. 1, pp. 37–42.
5. Mantawy A.H., Abdel-Magid Y.L., Selim S.Z. Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem. *IEEE Transactions on Power Systems*, 1999, vol. 14, iss. 3, pp. 829–836.
6. Matrenin P.V., Sekaev V.G. Sistemnoe opisanie algoritmov roevogo intellekta [Systems approach to swarm intelligence]. *Programmnaya inzheneriya – Software Engineering*, 2013, no. 12, pp. 39–45.
7. Matrenin P.V. Opisanie i realizatsiya algoritmov roevogo intellekta s ispol'zovaniem sistemnogo podkhoda [Description and implementation of Swarm Intelligence algorithms using the system approach]. *Programmnaya inzheneriya – Software Engineering*, 2015, no. 3, pp. 27–34.
8. Kennedy J., Eberhart R. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.
9. Matrenin P.V., Sekaev V.G. Particle swarm optimization with velocity restriction and evolutionary parameters selection for scheduling problem. *International Siberian Conference on*

- Control and Communications SIBCON 2015*: proceedings, Russia, Omsk, 21–23 May 2015, pp. 1–5.
10. Manusov V.Z., Matrenin P.V. Adaptivnyi algoritm roya chastits v zadachakh operativnogo planirovaniya [Adaptive particle swarm optimization for the operational scheduling problem]. *Vestnik komp'yuternykh i informatsionnykh tekhnologii – Herald of computer and information technologies*, 2016, no. 4 (142), pp. 11–15.
 11. Pham D.T., Ghanbarzadeh A., Коç E., Otri S., Rahim S., Zaidi M. The Bees Algorithm – a novel tool for complex optimisation problems. *IPROMS 2006. Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems*. Oxford, Elsevier, 2006, pp. 454–459. doi: <https://doi.org/10.1016/B978-008045157-2/50081-X>.
 12. Holland J.H. *Adaptation in natural and artificial systems*. Ann Arbor, University of Michigan Press, 1975. 225 p.
 13. Pedersen M., Chippereld A. Simplifying particle swarm optimization. *Applied Soft Computing*, 2010, no. 10 (2), pp. 618–628.
 14. Matrenin P.V., Sekaev V.G. Optimizatsiya adaptivnogo algoritma murav'inoi kolonii na primere zadachi kalendarnogo planirovaniya [Optimizing adaptive ant colony algorithm on the example of scheduling problem]. *Programmnaya inzheneriya – Software Engineering*, no. 4, pp. 34–40.
 15. Wolpert D.H., Macready W.G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, vol. 1, iss. 1, pp. 67–82.

СВЕДЕНИЯ ОБ АВТОРАХ



Манусов Вадим Зиновьевич – родился в 1936 году, д-р техн. наук, профессор, профессор кафедры систем электроснабжения предприятий Новосибирского государственного технического университета. Область научных интересов: методы искусственного интеллекта и вероятностные модели в энергетике. Опубликовано 200 научных работ. (Адрес: 630078, Россия, Новосибирск, Выставочная, 32/1. E-mail: manusov@mail.ru).

Manusov Vadim Zinovievich (b. 1936) – D. Sc. (Eng.), professor, department of power supply systems, Novosibirsk State Technical University. His research interests are currently focused on artificial intelligence and probabilistic methods in power engineering. He is the author of 200 scientific papers. (Address: 32/1, Vystavochnaya St., Novosibirsk, 630078, Russia. E-mail: manusov@mail.ru).



Матренин Павел Викторович – родился в 1991 году, аспирант кафедры систем электроснабжения предприятий Новосибирского государственного технического университета. Область научных интересов: стохастическая оптимизация, искусственный интеллект. Опубликовано 40 научных работ. (Адрес: 630084, Россия, Новосибирск, Д. Давыдова, 1. E-mail: pavel.matrenin@gmail.com).

Matrenin Pavel Viktorovich (b. 1991) – a postgraduate student, department of power supply system, Novosibirsk State Technical University. His research interests are currently focused on stochastic optimization methods and artificial intelligence. He is the author of 40 scientific papers. (Address: 1, D. Davydova St., Novosibirsk, 630084, Russia. E-mail: pavel.matrenin@gmail.com).

Статья поступила 05 сентября 2017 г.
Received September 05, 2017

To Reference:

Manusov V.Z., Matrenin P.V. Issledovanie dinamicheskikh svoystv populyatsionnykh algoritmov v zadache operativnogo upravleniya istochnikami reaktivnoi moshchnosti v sistemakh elektrosnabzheniya [Research on dynamic properties of population algorithms in operation control of reactive power units in smart grid]. *Doklady Akademii nauk vysshei shkoly Rossiiskoi Federatsii – Proceedings of the Russian higher school Academy of sciences*, 2017, no. 3 (36), pp. 74–87. doi: [10.17212/1727-2769-2017-3-74-87](https://doi.org/10.17212/1727-2769-2017-3-74-87)