

Алгоритм дифференциальной эволюции для оптимизации направленности фазированных антенных решеток

Еремеев А.В.¹, Тюнин Н.Н.¹

Институт математики им. С.Л. Соболева СО РАН, Омск, Россия,
eremeev@ofim.oscsbras.ru n.n.tyunin@gmail.com

Аннотация Задача оптимизации направленности фазированных антенных решеток коротковолнового диапазона сформулирована как невыпуклая задача квадратичного программирования. Метод дифференциальной эволюции адаптирован к специфике рассматриваемой задачи и проведен вычислительный эксперимент, в котором предложенный алгоритм сравнивается с методом градиентного подъема и пакетом BARON, основанном на градиентном подъеме и методе ветвей и границ. Проведен анализ наличия линейных симметрий поставленной задачи.

Keywords: Антенные решетки · квадратичное программирование · дифференциальная эволюция · группы симметрий · вычислительный эксперимент

Введение

В настоящее время разработка и анализ эффективных систем радиосвязи имеет большое значение для народного хозяйства. Одной из актуальных задач в этой области является задача оптимизации направленности фазированных антенных решеток (ФАР), представляющих собой антенные системы, распределение фаз и амплитуд на элементах которых позволяет получать направленное излучение. Будучи собранными в антенную систему и разведенными в пространстве, излучатели формируют диаграмму направленности, которая зависит от расположения и конструкции излучателей, а также выбора фаз и амплитуд сигналов, подаваемых на вход излучателей.

В диапазоне сверхвысоких частот (СВЧ) задачи оптимизации фаз и амплитуд излучателей, как правило, решаются с использованием некоторых упрощающих предположений [8,26,25]. Однако, в диапазоне высоких частот (ВЧ) задача оптимизации направленности ФАР оказывается более сложной, и потому менее изучена [24,28]. При ограничении суммарной мощности, подаваемой на антенную систему, задача выбора фаз и амплитуд на излучателях может быть решена аналитически [27]. Однако, при ограничении на мощность по каждому входу антенной системы требуется решение невыпуклых задач квадратичного программирования [6]. Вообще говоря, задачи квадратичного программирования являются NP-трудными [13]. Для решения таких задач могут применяться методы ветвей и границ [20], отсечений [7], ДС-программирования [21], полуопределенной релаксации [6], эволюционных вычислений [1,17], локального поиска [19] и др.

Данная работа посвящена исследованию свойств задачи оптимизации направленного излучения ФАР ВЧ диапазона и разработке алгоритмов решения этой задачи с использованием градиентного подъема и эволюционных вычислений. Результаты данной работы были кратко представлены в [3].

1 Основные обозначения

Математическая постановка задачи оптимизации направленности ФАР формулируется из физических соображений в виде выпуклого квадратичного функционала при невыпуклых

квадратичных ограничениях [27]:

$$\begin{cases} \mathbf{u}^+ \mathbf{A} \mathbf{u} \rightarrow \max, \\ 0 \leq \mathbf{u}^+ \mathbf{B}^{(1)} \mathbf{u} \leq, \\ \dots \\ 0 \leq \mathbf{u}^+ \mathbf{B}^{(N)} \mathbf{u} \leq 1. \\ \mathbf{u} \in \mathbb{C}^N \end{cases} \quad (1)$$

Здесь n – количество излучателей в системе. Матрица \mathbf{A} описывает влияние каждого элемента системы на ее излучение вдоль заданного направления. Ограничения, описываемые матрицами $\mathbf{B}_k, k = \overline{1, N}$ накладываются на мощность, которую генератор сигнала способен подать на каждый излучатель. Вектор \mathbf{u} представляет собой вектор-столбец комплексных напряжений. Исследование возможности оптимизация фаз и амплитуд этих величин и является целью данной работы.

Для построения алгоритмов анализа и решение данной задачи был предложен переход от постановки в комплексных числах к постановке в вещественных [22]:

$$\begin{cases} \mathbf{x}^T \mathbf{G} \mathbf{x} \rightarrow \max, \\ 0 \leq \mathbf{x}^T \mathbf{H}^{(1)} \mathbf{x} \leq 1, \\ \dots \\ 0 \leq \mathbf{x}^T \mathbf{H}^{(N)} \mathbf{x} \leq 1, \\ \mathbf{x} \in \mathbb{R}^{2n}. \end{cases} \quad (2)$$

Здесь матрица \mathbf{G} по смыслу и свойствам соответствует матрице \mathbf{A} , а матрицы $\mathbf{H}_k, k = \overline{1, N}$ – матрицам $\mathbf{B}_k, k = \overline{1, N}$.

Ранее было показано [22], что задача (1) имеет тривиальную симметрию $\mathbf{u} \rightarrow e^{j\phi} \mathbf{u}$, где j – мнимая единица, ϕ – фазовый сдвиг, одновременно применяемый на все компоненты вектора \mathbf{u} . При исследовании структуры локальных оптимумов в указанной работе было отмечено, что, возможно, задача (1) имеет и другие симметрии. В текущей работе ставится задача нахождения всех групп непрерывных симметрий задачи (1).

Также, в работе [22] для отыскания локального оптимума использовался многократный запуск градиентного подъема. Здесь для этих целей предлагается гибридный алгоритм, сочетающий элементы градиентного метода и дифференциальной эволюции.

При этом, задача (2) сводится к задаче безусловной оптимизации методом внешней точки в виде (3).

$$\mathbf{x}^T \mathbf{G} \mathbf{x} - r \cdot \sum_{k=1}^n \left(\min \left(0, \mathbf{x}^T \mathbf{H}^{(k)} \mathbf{x} \right) + \min \left(0, 1 - \mathbf{x}^T \mathbf{H}^{(k)} \mathbf{x} \right) \right)^\alpha \rightarrow \max, \quad (3)$$

где r и α – штрафные параметры.

2 Дифференциальная эволюция

2.1 Базовый вариант алгоритма

Эволюционные алгоритмы (ЭА) — один из наиболее широко используемых методов решения сложных задач оптимизации. Несколько вариантов этих стратегий были разработаны и применены во многих областях, таких как наука, экономика и инженерия. Среди них дифференциальная эволюция (ДЭ) [18] — одна из наиболее эффективных стратегий непрерывной оптимизации. Более того, ДЭ была признана стратегией-победителем нескольких конкурсов по оптимизации [2]. Подобно другим эволюционным алгоритмам, ДЭ вдохновлен естественным процессом эволюции и включает в себя применение мутаций, рекомбинаций и селекции.

Основная особенность метода ДЭ заключается в том, что он учитывает различия среди векторов, присутствующих в популяции, для изучения пространства поиска. В этом смысле он похож на оптимизаторы Nelder-Mead [14] и Controlled Random Search (CRS) [16].

ДЭ – стохастический алгоритм на основе популяции, поэтому он итеративно выводит несколько наборов решений-кандидатов. В ДЭ такие решения-кандидаты обычно называются векторами. В базовом варианте ДЭ для каждого члена популяции (их называют целевыми векторами) создается новый мутантный вектор. Затем мутантный вектор комбинируют с целевым вектором для создания пробного вектора. Наконец, применяется фаза селекции для выбора выживших. Таким образом, несколько поколений популяции эволюционируют, пока не будет достигнут критерий остановки. В поколении G i -й вектор популяции обозначается как $X_{i,G} = [x_{1,i,G}, x_{2,i,G}, \dots, x_{D,i,G}]$. Ниже приведены более подробные сведения о каждой фазе ДЭ.

Эксперименты показывают [15], что в целом эволюция популяции соответствует динамике случайного облака точек, движущегося как целое вдоль рельефа оптимизируемой функции, повторяя его характерные особенности. В случае попадания в овраг «облако» принимает форму этого оврага и распределение точек становится таким, что математическое ожидание разности двух случайных векторов оказывается направленным вдоль длинной стороны оврага. Это обеспечивает быстрое движение вдоль узких вытянутых оврагов, тогда как для градиентных методов в аналогичных условиях характерна колебательная динамика «от стенки к стенке». Приведенные эвристические соображения иллюстрируют наиболее важную и привлекательную особенность алгоритма ДЭ – способность динамически моделировать особенности рельефа оптимизируемой функции. Именно этим объясняется замечательная способность алгоритма быстро проходить сложные овраги, обеспечивая эффективность даже в случае сложного рельефа.

Инициализация. ДЭ обычно начинает процесс оптимизации со случайно инициированной популяции, состоящей из D векторов. Поскольку информация о перспективности различных регионов обычно отсутствует, применяются однородные генераторы случайных чисел: j -я компонента i -го вектора инициализируется как $x_{j,i,0} = a_j + rand_{i,j}[0, 1](b_j - a_j)$, где $rand_{i,j}[0, 1]$ – равномерно распределенное случайное число, лежащее между 0 и 1.

Мутация. Для каждого целевого вектора создается мутантный вектор. В настоящее время известно несколько способов выполнения этого процесса. В классическом варианте ДЭ применяется стратегия $rand/1$. В этом случае мутантный вектор $V_{i,G}$ создается следующим образом:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}), r_1 \neq r_2 \neq r_3 \quad (4)$$

Индексы $r_1, r_2, r_3 \in [1, D]$ – различные целые числа, случайно выбранные из диапазона $[1, D]$. Кроме того, все они отличаются от индекса i . Важно учитывать, что разница между векторами масштабируется числом F , которое называется силой мутации и обычно определяется в интервале $[0.4, 1]$.

Рекомбинация. Для объединения информации о различных решениях-кандидатах и с целью увеличения разнообразия применяется оператор кроссинговера. В частности, каждый целевой вектор $X_{i,G}$ смешивается с соответствующим ему мутантным вектором $V_{i,G}$ для создания пробного вектора $U_{i,G} = [u_{1,i,G}, u_{2,i,G}, \dots, u_{D,i,G}]$. Наиболее типичным кроссовером является биномиальный, который работает следующим образом:

$$U_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{если } rand_{i,j}[0, 1] \leq CR \text{ или } j = j_{rand} \\ x_{j,i,G}, & \text{иначе} \end{cases}, \quad (5)$$

где $rand_{i,j}[0, 1]$ – равномерно распределенное случайное число, j_{rand} – случайно выбранный индекс, который гарантирует, что $U_{i,G}$ наследует хотя бы одну компоненту от $V_{i,G}$, $CR \in [0, 1]$ – интенсивность кроссовера.

Селекция. Наконец, выполняется жадный отбор для определения оставшихся в живых следующего поколения. Каждый пробный вектор сравнивается с соответствующим ему целевым вектором, и выживает лучший из них:

$$X_{i,G+1} = \begin{cases} U_{i,G}, & \text{если } \tilde{F}(U_{i,G}) \leq \tilde{F}(X_{i,G}) \\ X_{i,G}, & \text{иначе.} \end{cases} \quad (6)$$

Следовательно, каждый член популяции либо становится лучше по целевой функции, либо остается с тем же значением целевой функции в следующем поколении.

2.2 Гибридный вариант алгоритма

Будучи примененной к различным примерам из [22], описанный выше базовый вариант алгоритма дифференциальной эволюции зачастую не мог обнаружить даже допустимых решений за все отведенное ему время. Чтобы избежать этой тенденции, в данной работе предложен гибридный вариант ДЭ с использованием градиентного метода, масштабированием решений в допустимую область и адаптацией штрафа, учитывающий специфику решаемой задачи.

Гибридная модификация может быть описана следующей общей схемой:

```

i := номер текущей итерации
i0 := номер итерации, на которой было получено улучшение рекорда
D := размер популяции
Grad := градиентный алгоритм
X := лучшая особь популяции
если i > D И i > 2 * i0 то {
    Xnext := Grad(X)
    если Xnext = X то завершить выполнение, X является решением
    иначе {
        X = Xnext
        i0 = i
    }
}

```

Условие $i > 2 * i_0$ определяет, происходило ли улучшение популяции за определенное количество итераций. Условие $i > D$ гарантирует, что на начальных итерациях ДЭ не будет принято решение о запуске градиентного алгоритма [4].

Для рассматриваемой задачи существует преобразование, позволяющее привести к допустимой области решение x , которое нарушает только ограничивающие неравенства задачи (3) вида $x^T H^{(k)} x \leq 1$:

$$x' := \alpha(x)^{-1/2} x, \quad (7)$$

где $\alpha(x) := \max_{k=1, \dots, n} x^T H^{(k)} x$. Поскольку как целевая функция, так и ограничения представлены квадратичными формами, применение такой операции приведет к пропорциональному уменьшению в $\alpha(x)$ раз значений каждой из квадратичных форм. Другими словами, если в некоторой точке x значения каждой из квадратичных форм, задающих ограничения, больше 0, причем значения некоторых из них больше 1, то по формуле (7) можно определить множитель, умножение которого на вектор x ведет к тому, что наибольшее из значений квадратичных форм, задающих ограничения, будет равно 1. Данная процедура применяется для выбора начального решения, а также для масштабирования каждого нового решения перед оценкой его качества в алгоритме ДЭ. При этом если решение до масштабирования имело большие нарушения ограничений, то после масштабирования его целевая функция существенно изменяется, как правило в сторону снижения качества. На этом принципе основано адаптивное правило подбора штрафного коэффициента, описываемое далее.

Кроме того, для улучшения качества решений в гибридном варианте предлагается динамическое увеличение штрафного коэффициента:

```

 $i$  := номер текущей итерации
 $i_0$  := номер итерации, на которой было получено улучшение рекорда
 $i_1$  := номер итерации, на которой произошло предыдущее увеличение штрафа
 $D$  := размер популяции
 $r$  := штрафной коэффициент
если  $i > D$  И  $i > 1.5 * i_0$  И  $i > 2 * i_1$  то {
     $r := 2r$ 
     $i_1 := i$ 
}

```

Здесь условие $i > D$ гарантирует, что на начальных итерациях не будет принято решение об увеличении штрафа, а условие $i > 1.5 * i_0$ - что лучшее значение популяции не было улучшено за определенное количество итераций. Условие $i > 2 * i_1$ «замедляет» следующее срабатывание критерия. Таким образом, при отсутствии существенных улучшений качества решений происходит увеличение штрафного коэффициента. С учетом масштабирования особей в допустимую область, это приводит к выживанию особей с меньшим нарушением ограничений и позволяет улучшить качество получаемых решений.

2.3 Вычислительный эксперимент

Вычислительный эксперимент был поставлен для задач, рассмотренных в [22,23]. В данной статье ШВИК, ШВДК, СВДК обозначают решетки кольцевой структуры. Далее следует указание количества элементов (8 или 16). Через тире – расстояние от центра излучателя до центра решетки (15, 20, 25, 30, 37). Для ШВД приводится плотность противовесов (например, 3:3). Поскольку современные ЭВМ на аппаратном уровне поддерживают вычисление в параллельном режиме, рассматриваемый здесь алгоритм ДЭ был адаптирован под эту особенность: за один запуск алгоритма производится 4 параллельных выполнения, а на выход подается лучшее решение. Это позволяет использовать возможности современных ЭВМ для получения более качественных решений. В таблице 1 результаты, полученные с помощью ДЭ, сравниваются с результатами коммерческого решателя BARON и в пакете GAMS. Решатель BARON, также как ДЭ, имеют ограничение по времени счета 1000с (эта длительность выбрана из практических соображений, т.к. она сопоставима с временем построения исходных данных с использованием системы NEC, и совпадает с ограничением по времени, выбранным в [22]). В случае нескольких запусков ДЭ за отведенное время, за решение принимается лучшее из найденных за 1000с. Для оценки среднего значения целевой функции на выходе ДЭ производилась серия из 10 независимых испытаний по 1000с в каждом. Результаты эксперимента при выборе типичных значений настраиваемых параметров ДЭ (популяция из 100 особей, $F = 0.6$, $CR = 0.6$) представлены в таблице 1. Значения целевой функции в таблице округлены до целых. Полужирным шрифтом выделены случаи, когда указанное значение целевой функции не менее, чем на 1% выше, чем у другого алгоритма. Вычисления производились на ЭВМ с процессором Intel i7 (тактовая частота: 2.8ГГц), ОЗУ: 16Гб.

Аналогичные испытания производились и с помощью решателя ANTIGONE в пакете GAMS, однако, было выявлено, что в режиме по умолчанию данный решатель на всех задачах выдал нулевое решение, кроме СВД 2х2, где решение по целевой функции совпадало с результатом BARON. Также, в результате проведенных экспериментов было обнаружено, что динамическая адаптация штрафного коэффициента позволяет ДЭ достичь более качественных решений, что особенно заметно в тех случаях, когда результаты ДЭ и BARON близки по качеству.

С целью изучения возможности ускорения работы решателей за счет учета специфики задачи были проведены дополнительные исследования структуры рассматриваемых примеров с точки зрения линейных симметрий этих задач [5]. Ранее было отмечено (см. [22]), что решения рассматриваемой задачи эквивалентны с точностью до сдвига фаз во всех излучателях на равную величину. Учет данной симметрии (для краткости называемой «фазовой симметрией») может быть реализован фиксацией в ноль одной из переменных задачи, например, $x_1 = 0$. В результате добавления такого ограничения к условиям задачи число переменных

Таблица 1: Результаты экспериментов, полученные с помощью ДЭ и BARON

Тип	ДЭ	BARON	
	\tilde{F}	\tilde{F}	t, c
ШВИ 2x2	139	139	0.12
ШВИ 3x3	580	580	0.34
ШВД 2x2	463	463	0.27
ШВД 3x3	924	925	0.34
СВД 2x2	361	361	0.16
СВД 3x3	1163	1261	0.38
СВД 5x5	7132	6716	1000
СВД' 2x2	198	253	0.25
СВД' 3x3	834	1153	1.4
СВД' 5x5	2755	33	217.94
ШВИК 8-15(3:3)	218	218	0.23
ШВИК 16-15(3:7)	732	734	1.37
ШВИК 8-15(2:3)	1664	-	14.62
ШВДК 8-20	1454	1454	2.78
ШВДК 8-30	2421	2421	1.47
СВДК 8-25	732	734	0.23
СВДК 8-37	1486	1486	0.23

сокращается на единицу и можно предположить, что это сократит время вычислений для известных алгоритмов. Нами поставлены вопросы о том, действительно ли происходит сокращение времени вычислений и о существовании других семейств симметрий, которые могли бы еще более сократить пространство поиска решаемой задачи.

Для ответа на первый вопрос на всех тестовых примерах был найден коэффициент ускорения решателя BARON, получаемый от фиксации $x_1 = 0$. Результаты представлены на рисунке 1 (здесь отсутствует задача СВД' 5x5, где алгоритм с фиксацией затратил существенно большее время, но при этом нашел решение с большим на 10% значением целевой функции, а также задача ШВИК 8-15(2:3), в которой решения не были найдены в обоих случаях). В большинстве примеров фиксация переменной привела к ускорению работы алгоритма, среднее ускорение по представленным здесь задачам составило 0.95, что говорит о целесообразности фиксации в ноль одной из переменных при использовании решателя. Аналогичный эксперимент с алгоритмом ДЭ и решателем ANTIGONE не показал существенного улучшения качества решений или их ускорения в результате фиксации x_1 .

Для ответа на второй вопрос был проведен поиск непрерывной группы линейных симметрий в рассматриваемых примерах, как описано ниже в приложении. В результате было установлено, что других непрерывных семейств линейных симметрий в рассматриваемых примерах не существует. Вопрос о поиске дискретных симметрий пока остается открытым.

Из проведенных экспериментов можно сделать следующие выводы:

- Разработанный в рамках данной работы гибридный вариант ДЭ показывает конкурентоспособные результаты по сравнению с коммерческим решателем BARON в режиме его настроек по умолчанию, при этом преимущество ДЭ наблюдается на задачах с наибольшей размерностью (25 и 16 переменных).
- Решатель ANTIGONE в режиме его настроек по умолчанию в большинстве тестовых примеров возвращает нулевое решение.
- Учет фазовых симметрий задачи (1), как правило, позволяет ускорить работу решателя BARON.
- Других непрерывных семейств линейных симметрий в рассматриваемых задачах не существует.

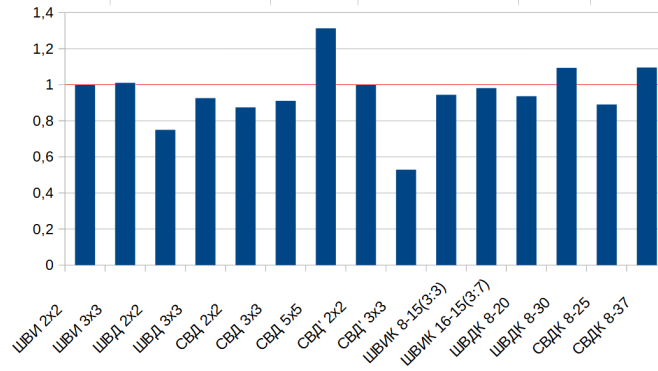


Рис.1: Отношение длительности вычислений с фиксацией первой координаты к исходной длительности вычислений

3 Заключение

В рамках данной работы был разработан гибридный вариант алгоритма дифференциальной эволюции с использованием градиентного алгоритма и адаптацией штрафа. Показано, что разработанный вариант ДЭ демонстрирует конкурентоспособные результаты.

В целях ускорения работы алгоритмов для рассмотренных задач была применена методика анализа наличия группы непрерывных симметрий, в результате которой было обнаружено наличие только фазовой симметрии. Было произведено сравнение гибридного алгоритма и коммерческих решателей с учетом обнаруженной симметрии и без нее. Выявлено, что учет симметрии может привести к ускорению работы решателя BARON.

Приложение

Линейные симметрии задачи квадратичного программирования

Решение и анализ задач математического программирования могут быть упрощены при наличии симметрий этих задач, соответствующих некоторым линейным преобразованиям. В частности, знание таких симметрий может быть использовано для уменьшения размерности задачи, ограничения пространства поиска или получения нового локального оптимума из имеющегося. В настоящей работе исследуется случай непрерывной области решений. В то время как предыдущие исследования симметрии в математическом программировании обычно имели дело с перестановками координат пространства решений [9,10,12], здесь рассматривается большая группа обратимых линейных преобразований. Мы изучаем частный случай задачи квадратичного программирования с квадратичными ограничениями в \mathbb{R}^N : целевая функция и ограничения задаются квадратичными формами G , и H_1, \dots, H_n , в виде (2). Следует отметить, что все матрицы G , и H_1, \dots, H_n , симметричны и $H = \sum_{i=1}^n H_i$ положительно определена. Без потери общности будем считать, что все ограничения заданы неравенствами \leq .

Под симметрией задачи (2) подразумевается линейное преобразование

$$x \rightarrow y = Px, \quad (8)$$

определенное невырожденной матрицей P , такой что задача (2), выраженная в терминах преобразованного пространства (т.е. через вектор-столбец y), совпадает с исходной задачей. Таким образом, в терминах вектора y задача (2) формулируется в той же форме.

$$\begin{cases} y^T G y \rightarrow \max, \\ y^T H_1 y \leq 1, \\ \dots \\ y^T H_N y \leq 1, \end{cases} \quad (9)$$

с той же самой матрицей G и тем же набором матриц $\{H_k, k = \overline{1, N}\}$.

В [5] показано, что симметричная матрица $H = \sum_{k=1}^n H_k, k = \overline{1, N}$ может быть представлена как конгруэнтное преобразование диагональной матрицы:

$$H = S^T D S, \quad (10)$$

где D – диагональная матрица, которая может иметь только “0”, “1”, или “-1” на ее главной диагонали. Матрица S может быть получена конструктивно, например, конечным методом Лагранжа ([11], Г. 5).

Матрица S описывает переход в группу ортогональных преобразований, в которой производится поиск подгрупп непрерывных симметрий. Любой ее элемент такой группы может быть выражен с помощью базисных элементов, называемых генераторами:

$$X = \sum_k a_k G_k, \quad (11)$$

где a_k являются вещественными числами, G_k являются генераторами. Пространство кососимметричных матриц имеет размерность $N(N-1)/2$, а количество коэффициентов a_k будет равно количеству генераторов. В качестве генераторов можно выбрать матрицы, у которых над главной диагональю все элементы равны 0, кроме одного элемента, равного 1. Тогда кососимметрия однозначно определяет остальные матричные элементы этих генераторов. Итак, любой элемент Q из $SO(N)$ можно представить в виде:

$$Q = e^{\sum_k a_k G_k}. \quad (12)$$

Сам поиск осуществляется решением системы линейных уравнений (13):

$$\begin{cases} \tilde{H}_n \left(\sum_k a_k G_k \right) = \left(\sum_k a_k G_k \right) \tilde{H}_n, \\ \tilde{G} \left(\sum_k a_k G_k \right) = \left(\sum_k a_k G_k \right) \tilde{G}. \end{cases} \quad (13)$$

Уравнения (13) представляют собой систему линейных алгебраических уравнений, определяющих параметры a_k . Эта система однородна, поэтому она имеет континуум ненулевых решений или одно тривиальное решение. Тривиальное нулевое решение всегда присутствует и соответствует единичной матрице Q . Некоторые из параметров a_k остаются “свободными” (это будут параметры искомой подгруппы), а остальные из a_k могут быть линейно выражены через “свободные”. Решение этой системы уравнений (13) может быть получено конструктивно методом Гаусса.

Условие инвариантности задачи относительно преобразования Q превращается в

$$Q = e^{\sum_k a_k \hat{G}_k}, \quad (14)$$

где сумма идет по «свободным» параметрам a_k , а новые генераторы, обозначаемые через \hat{G}_k , являются линейными комбинациями прежних генераторов G_k . Множество всех Q -матриц, удовлетворяющих (14), параметризуется конечным набором вещественных параметров a_k .

Поиск группы непрерывных симметрий в задаче оптимизации направленности фазированных антенных решеток

Вычисления состояли из следующих этапов:

1. Обработка. На этом этапе возможная неточность данных нивелируется усреднением симметричных компонент матриц (матрицы G и H должны быть симметричны).

2. Преобразование $\mathbf{H}_\Sigma = \sum_i \mathbf{H}_i$ к канонической форме используя метод Лагранжа для вычисления матриц \mathbf{S} и \mathbf{S}^{-1} .
3. Применение метода Гаусса к системе линейных уравнений (13) для вычисления генераторов \hat{G}_n .

Следует отметить, что входные данные могут содержать некоторые погрешности, которые приводят к несимметричности матриц \mathbf{G} и \mathbf{H} , что может существенно повлиять на поиск непрерывных групп симметрий. Таким образом, на этапе 1, мы используем известные свойства задачи чтобы нивелировать влияние погрешности. Также, в методе Гаусса на шаге 3, любые значения принимаются за 0 если их абсолютное значение меньше определенного порогового значения Δ , который является параметром алгоритма. Причина в том, что последовательное исключение переменных из уравнений, выполняемое методом Лагранжа с представлением вещественных чисел с плавающей запятой, не может гарантировать идеальную точность. В результате некоторые линейно зависимые строки матрицы не могут быть исключены, что может привести к неверному результату. Большое значение порога Δ может привести к вырожденности задачи, тогда как слишком малое значение Δ не позволит выявить линейные зависимости.

При вычислениях Δ изменялось от 10^{-4} до 10^{-12} . В данном диапазоне для каждого рассмотренного частного случая задачи, не было получено различий в полученных решениях.

Описанная процедура нахождения непрерывных групп симметрий применяется к примерам, описанным в [22]. Для всех рассмотренных задач было выявлено только наличие симметрии относительно преобразования $i \rightarrow e^{j\phi}i$ всех комплексных координат (по произвольному углу ϕ). За j здесь обозначена мнимая единица. Данная симметрия находит применение для уменьшения размерности области поиска на единицу. Например, фиксируя $\text{Im}(y_N) = 0$, что эквивалентно добавлению ограничения $x_{2N} = 0$ к задаче (2). Возможно, множественность решений объясняется наличием дискретных симметрий. Выявление дискретных симметрий является объектом дальнейших исследований.

В результате работы алгоритма было выявлено, что все генераторы \hat{G}_k могут быть выражены через один новый генератор вида

$$\hat{G} = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

который соответствует фазовой симметрии. Для данного генератора при $a = 1$

$$e^{a\hat{G}} = \begin{pmatrix} 0.5403 & 0 & 0 & 0 & -0.8415 & 0 & 0 & 0 \\ 0 & 0.5403 & 0 & 0 & 0 & -0.8415 & 0 & 0 \\ 0 & 0 & 0.5403 & 0 & 0 & 0 & -0.8415 & 0 \\ 0 & 0 & 0 & 0.5403 & 0 & 0 & 0 & -0.8415 \\ 0.8415 & 0 & 0 & 0 & 0.5403 & 0 & 0 & 0 \\ 0 & 0.8415 & 0 & 0 & 0 & 0.5403 & 0 & 0 \\ 0 & 0 & 0.8415 & 0 & 0 & 0 & 0.5403 & 0 \\ 0 & 0 & 0 & 0.8415 & 0 & 0 & 0 & 0.5403 \end{pmatrix}.$$

Список литературы

1. Boriskin, A. V., Balaban, M. V., Galan, O. Yu., Sauleau, R.: Efficient approach for fast synthesis of phased arrays with the aid of a hybrid genetic algorithm and a smart feed representation. 2010 IEEE International Symposium on Phased Array Systems and Technology pp. 827–832 (2010)

2. Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation* 15(1), 4–31 (2011)
3. Ereemeev, A., Tiunin, N.: Differential evolution for short wave antenna array optimization. *motor 2022: Mathematical optimization theory and operations research* : [сайт]. – 2022., <https://easychair.org/smart-program/MOTOR2022/2022-07-04.html#talk:195757>
4. Ereemeev, A. V.: A restarting rule based on the schnabel census for genetic algorithms. Battiti, R., Brunato, M., Kotsireas, I., Pardalos, P. (eds) *Learning and Intelligent Optimization. LION 12 2018. Lecture Notes in Computer Science* 11353 (2018)
5. Ereemeev, A. V., Yurkov, A. S.: On symmetry groups of some quadratic programming problems. *Mathematical Optimization Theory and Operations Research. Proc. 19th Int. Conf. MOTOR 2020 (Novosibirsk, Russia, July 6–10, 2020). Springer, LNCS* 12095, 35–48 (2020)
6. Fuchs, B.: Application of convex relaxation to array synthesis problems. *IEEE Transactions on Antennas and Propagation* 62(2), 634–640 (2014)
7. Horst, R., Pardalos, P. M.: *Handbook of global optimization*, vol. 2. Springer Science & Business Media (2013)
8. Indenbom, M., Izhutkin, V., Sharapov, A., Zonov, A.: Synthesis of conical phased antenna arrays optimization of amplitude distribution parameters. *IX International Conference on Optimization and Applications (OPTIMA 2018)* pp. 273–285 (2018)
9. Kolokolov, A. A., Orlovskaya, T. G., Rybalka, M. F.: Analysis of integer programming algorithms with l-partition and unimodular transformations. *Automation and Remote Control* 73(2), 369–380 (2012)
10. Kouyialis, Georgia., Wang, Xiaoyu., Misener, Ruth.: Symmetry detection for quadratic optimization using binary layered graphs. *Processes* 7(11) (2019)
11. Lancaster, Peter., Tismenetsky, Miron.: *The Theory of Matrices*. Academic Press (1985)
12. Liberti, Leo.: Reformulations in mathematical programming: automatic symmetry detection and exploitation. *Mathematical Programming* 131 (2012)
13. Murty, K.: Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming. North Holland* 39, 117–129 (1987)
14. Nelder, J.A., Mead, R.: A simplex method for function minimization. *The computer journal* 7(4), 308–311 (1965)
15. Price, K. V., Storn, R. M., Lampinen, J. A.: *Differential Evolution. A Practical Approach to Global Optimization*. Springer (2005)
16. Price, W.: *Journal of optimization theory and applications*. *The computer journal* 40(3), 333–348 (1983)
17. Rao, A., Sarma, N.: Synthesis of reconfigurable antenna array using differential evolution algorithm. *IETE Journal of Research* 63(3), 428–434 (2017)
18. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*. Springer 11(4), 341–359 (1997)
19. Кочетов, Ю. А.: Вычислительные возможности локального поиска в комбинаторной оптимизации. *Ж. вычисл. матем. и матем. физ.* 48(5), 747–763 (2008)
20. Нечаева, М. С., Хамисов, О. В.: Метод ветвей и границ для задачи минимизации невыпуклой квадратичной функции при выпуклых квадратичных ограничениях. *Дискретн. анализ и исслед. опер.* 7(2), 74–88 (2000)
21. Стрекаловский, А. С.: О минимизации разности выпуклых функций на допустимом множестве. *Ж. вычисл. матем. и матем. физ.* 43(3), 399–409 (2003)
22. Тюнин, Н.Н.: Задачи невыпуклого квадратичного программирования, связанные с оптимизацией фазированных антенных решёток. *ДАИО* 28(3), 65–89 (2021)
23. Тюнин, Н.Н.: Об оптимизации направленности коротковолновых фазированных антенных решеток кольцевой структуры. *Техника радиосвязи* (2022)
24. Фаняев, И. А., Кудин, В. П.: Фазированная антенная решетка кругового обзора над проводящей цилиндрической поверхностью из излучателей вертикальной поляризации. *Известия Гомельского государственного университета имени Ф. Скорины. Сер.: Естественные науки*. pp. 191–198 (2014)
25. Фаняев, И. А., Кудин, В. П.: Синтез амплитудного распределения на входах излучателей фазированной антенной решетки над цилиндрической поверхностью методом роя частиц. *Доклады БГУИР* pp. 89–96 (2017)
26. Щелкунов, С. А., Фрис, Г.: *Антенны: Теория и практика*. М.:Советское радио (1955)
27. Юрков, А. С.: Оптимизация возбуждения передающих фазированных антенных решеток декаметрового диапазона длин волн. *ОНИИП* (2014)
28. Юрков, А. С.: Максимизация направленности фазированных антенных решеток коротковолнового диапазона. *Техника радиосвязи. ОНИИП* (2), 46–53 (2016)