# МЕТОДЫ РАСПРОСТРАНЕНИЯ ОГРАНИЧЕНИЙ В РЕШЕНИИ НЕЛИНЕЙНЫХ УРАВНЕНИЙ И ИХ НЕКОТОРЫЕ ПРИЛОЖЕНИЯ

**Тохиров Ф.Ж.**

*PhD, Исполняющий обязанности доцента кафедры «Цифровые технологии» НавГУ*

**Аннотация:** В работе проводится разграничение между задачей удовлетворения ограничений (Constraint Satisfaction, CS) и процессом распространения ограничений (CP), которые представляют собой разные этапы решения. Подробно рассматривается мощный метод интервального распространения ограничений (Interval Constraint Propagation, ICP), предназначенный для работы с непрерывными доменами. Метод ICP сочетает классическое распространение ограничений с интервальной арифметикой, что позволяет эффективно обрабатывать нелинейные уравнения и неравенства, обеспечивая устойчивость к ошибкам округления. Для повышения эффективности ICP комбинируется с алгоритмом «разбиение и усечение» (branch-and-prune). Теоретические положения иллюстрируются подробными численными примерами, демонстрирующими пошаговую работу алгоритмов и их применение для поиска решений систем нелинейных уравнений с гарантированной точностью.

**Ключевые слова:** распространение ограничений, интервальные методы, нелинейные уравнения, задача удовлетворения ограничений, алгоритм branch-and-prune.

**Распространение ограничений** (также иногда называемое удовлетворение ограничений или программирование в ограничениях) представляет собой одну из самых динамично развивающихся областей искусственного интеллекта, ориентированную на решение разнообразных практических задач.

В настоящее время методы распространения ограничений активно применяются как фундамент для решения прикладных задач в различных сферах, например, при планировании ресурсов и составлении расписаний, временном анализе в системах принятия решений, математическом и инженерном моделировании, в задачах теории графов и других сложных вычислительных проблемах. Благодаря своей универсальности и эффективности, это направление вызывает значительный исследовательский интерес. На сегодняшний день в данной области уже достигнуты существенные теоретические результаты и созданы практические инструменты, успешно применяемые для решения широкого круга реальных задач.

В методах распространения ограничений выделяют два класса методов — методы для задач с дискретными переменными, называемые распространением ограничений над дискретными областями [1,2], и методы для задач с вещественными переменными, называемые распространением ограничений над непрерывными областями [3,4,5].

В настоящее время методы распространения ограничений все чаще объединяются с другими подходами, порождая новые эффективные алгоритмы [6, 7, 8]. Например, распространение ограничений над конечными областями используется в комбинации с методами исследования операций и линейного программирования, а распространение ограничений над непрерывными областями использует методы интервальной математики.

Говоря неформально, задача удовлетворения ограничений — это задача, состоящая из конечного числа переменных, каждая из которых связана с некоторой областью ее определения (дискретной или непрерывной), и конечного множества ограничений, которые ограничивают множества значений переменных, принимаемых одновременно. Решение задачи удовлетворения ограничений состоит в присваивании значения каждой переменной так, чтобы одновременно удовлетворялись все ограничения. Целью может являться как нахождение одного такого набора (любого или с заданным свойством), так и нахождение всех наборов.

Формально задача удовлетворения ограничений (CSP, Constraint Satisfaction Problem) определяется как тройка $\langle X, D, C \rangle$, где $X = \{x_1, x_2, \ldots, x_n\}$ — множество переменных, $D = D_1 \times D_2 \times \ldots \times D_n$ ($D_i$ — множество значений переменной $x_i$), $C$ — множество ограничений $\{C_1, C_2, \ldots, C_n\}$, $C_j$ — отношение (уравнение, неравенство, таблица), которое связывает некоторые переменные $x_1, x_2, \ldots, x_{n_j}$.

Хотя оба понятия, т.е. удовлетворения ограничений (Constraint Satisfaction, CS) и распространения ограничений (Constraint Propagation, CP) связаны с задачами ограничений, они относятся к разным этапам процесса решения задачи. Сопоставляем их по некоторым соответствующим критериям в следующей таблице (Таблица 1).

Таблица 1. Сопоставления задачи CS и CP по основным критериям.

| Критерий | Удовлетворение ограничений (CS) | Распространение ограничений (CP) |
|---|---|---|
| **Определение** | Общий процесс поиска решения, удовлетворяющего всем ограничениям. | Локальный процесс сужения доменов переменных на основе ограничений. |
| **Цель** | Найти полное решение (или все решения) задачи. | Уменьшить пространство поиска, отсекая заведомо неверные значения. |
| **Когда применяется?** | На этапе глобального поиска (например, backtracking). | На этапе предобработки или во время поиска для ускорения. |
| **Примеры методов** | Backtracking, branch-and-bound, SMT-решатели. | AC-3 (арковая согласованность), bound consistency, ICP (интервальный метод). |
| **Аналог** | Поиск маршрута в лабиринте. | Устранение тупиковых ветвей в лабиринте до начала поиска. |

При работе с непрерывными переменными очень удобно представление множества значений переменных с помощью интервалов (оно может быть также использовано и для целочисленных переменных). Итак, пусть $FP$ (Float Point) — множество чисел с плавающей точкой (чисел, представимых в компьютере), расширенное двумя элементами $\{-\infty, +\infty\}$. Тогда любое подмножество множества вещественных чисел может быть представлено минимальным интервалом $I=[a,b]$, где $a \in FP$, $b \in FP$, таким, что $X \subseteq I$. Методы решения численных задач, которые используют технику распространения ограничений при интервальном представлении множества значений переменных, называются методами интервального распространения ограничений.

**Интервальные методы распространения ограничений (Interval Constraint Propagation, ICP)** — это важное направление в теории удовлетворения ограничений, особенно для задач с непрерывными доменами (вещественными числами). Они сочетают классическое распространение ограничений с **интервальной арифметикой**, позволяя эффективно обрабатывать нелинейные уравнения и неравенства. Основная идея этого направления состоит в том, что интервальные методы распространения ограничений работают с **интервалами** (а не дискретными значениями), сужая их на основе ограничений. При этом каждая переменная $x$ представляется некоторым интервалом $[a,b]$, где $a,b \in \mathbf{R}$, а ограничения вида $f(x_1, x_2, \ldots, x_n) = 0$ или

$f(x_1, x_2, \dots, x_n) \le 0$ обрабатываются с помощью интервальной арифметики. Далее, для интервального оценивания решения на каждом шаге выбранный алгоритм пытается уменьшить интервалы переменных, отсекая значения, которые не удовлетворяют ограничениям.

Алгоритм ICP последовательно применяет два шага:

1. Вычисление **интервалов** для всех выражений в ограничениях.
2. **Сужение интервалов** переменных на основе ограничений.

Приводим псевдокод алгоритма ICP (Листинг 1):

**Листинг 1.** Алгоритм ICP.

```
Пока есть сужение интервалов:
  Для каждого ограничения C:
    Вычислить интервал f(X) с помощью интервальной арифметики.
    Если C имеет вид f(X)=0:
      Пересечь интервал f(X) с [0,0] → если пусто, нет решений.
      Скорректировать интервалы переменных из X.
    Если C имеет вид f(X)≤0:
      Пересечь f(X) с (−∞,0].
```

**Процесс повторяется до стабилизации интервалов.** Рассмотрим пример для работы алгоритма ICP (Пример 1).

**Пример 1.** Пусть дано $x \in [0,10]$, $y \in [0,10]$, с ограничениями $x + y = 5$ и $x - y \ge 1$.

***Шаг 1:*** Из $x + y = 5$ находим $y = 5 - x$, затем подставляем в $D_y$ и получаем $y \in [5-10,\ 5-0] = [-5,5]$. Но изначально $y \in [0,10]$, то пересечение дает $y \in [0,5]$.

***Шаг 2:*** Из $x - y \ge 1$ выразим $x \ge y + 1$, так как $y \in [0,\ 5]$, то $x \in [1,\ 10]$ и имеем $x \ge 0 + 1 = 1$.

***Шаг 3:*** Уточняем $y = 5 - x$: $x \in [1,\ 10] \Rightarrow y \in [5-10,\ 5-1] = [-5,\ 4]$. Находим пересечение $y \in [-5,\ 4] \cap [0,\ 5] = [0,\ 4]$.

***Шаг 4:*** Возвращаемся к $x \ge y + 1$: $y \in [0,4] \Rightarrow x \in [1,5]$ (так как $x \le 5$ из $x + y = 5$).

Таким образом, мы находили интервальные оценки: $x \in [1,\ 5]$ и $y \in [0,\ 4]$. Дальнейшие сужение требует разбиения интервалов, например, можно применять метод разбиение и усечение (branch-and-prune) [9].

**Листинг 2.** Псевдокод комбинации Branch-and-Prune и ICP.

```
def solve(X, Y):
   if is_empty(X) or is_empty(Y):
      return
   if is_small_enough(X, Y):  # Интервалы достаточно малы
      print(f"Решение: x∈X, y∈Y")
```

```
    return
  X1, X2 = split(X)  # Разбиваем X
  for X_sub in [X1, X2]:
    Y_new = propagate_constraints(X_sub, Y)  # ICP
    solve(X_sub, Y_new)
```

В следующем примере покажем применение алгоритма ICP в комбинации методом "branch-and-prune" для решения систем нелинейных уравнений (Пример 2, напомним, что интервальные величины выделяются жирным шрифтом [10]).

**Пример 2.** Пусть дана система, требуется найти решения $(x, y)$ в заданной области $x \in [-10,10] = \boldsymbol{x},\ \ y \in [-10,10] = \boldsymbol{y}$:

$$\begin{cases} x^2 + y = 4 \\ e^x + y = 5 \end{cases}.$$

***Шаг 1:*** Применяем алгоритм ICP.

**1.1.** Первую уравнению выразим как $y = 4 - x^2$, применяя операции интервальной арифметики [10] вычислим интервал $\boldsymbol{y}$: $x^2 \in [0, 100]$, тогда $y \in 4 - [0,100] = [-96, 4]$. Находим пересечение (т.е. сужаем $\boldsymbol{y}$) $\boldsymbol{y} = [-10, 10] \cap [-96, 4] = [-10, 4]$.

**1.2.** Обрабатываем второе уравнение: Находим $y = 5 - e^x$ и вычислим $e^x \in \left[e^{-10}, e^{10}\right] \approx [0, 22026]$. Тогда $y \in 5 - [0, 22026] = [-22021, 5]$. Текущий $\boldsymbol{y} = [-10, 4]$, сужаем $\boldsymbol{y} = [-10, 4] \cap [-22021, 5] = [-10, 4]$. Получаем интервальные оценки по ICP: $\boldsymbol{x} = [-10, 10],\ \ \boldsymbol{y} = [-10, 4]$.

***Шаг 2:*** Применяем алгоритм Branch-and-Prune. Для этого делим $\boldsymbol{x} = [-10, 10]$ на два подынтервала: $\boldsymbol{x}_1 = [-10, 0]$ и $\boldsymbol{x}_2 = [0, 10]$.

**2.1.** Обрабатываем $\boldsymbol{x}_1 = [-10, 0]$. Для первого уравнения $y = 4 - x^2$ на интервале $\boldsymbol{x}_1 = [-10, 0]$ находим $\boldsymbol{y} = 4 - \boldsymbol{x}^2 = [4, 4] - [-10, 0]^2 = [-96, 4]$. Новый $\boldsymbol{y} = [-10, 4] \cap [-96, 4] = [-10, 4]$ (без изменений). Для второго уравнения $y = 5 - e^x$ вычислим, что $e^x \in \left[e^{-10}, e^0\right] \approx [0, 1]$. Тогда $\boldsymbol{y} = 5 - e^x = [5, 5] - [0, 1] = [4, 5]$. Находим пересечение с исходным интервалом и получим $\boldsymbol{y} = [-10, 4] \cap [4, 5] = [4, 4] = 4$. Теперь подставляем найденное $y = 4$ в первое уравнение: $x^2 + 4 = 4\ \Rightarrow\ x^2 = 0\ \Rightarrow x = 0$. Таким образом, найдено решение: $x = 0,\ \ y = 4$. Здесь мы прекращаем рассмотрение интервала $\boldsymbol{x}_1 = [-10, 0]$.

**2.2.** Обрабатываем $\boldsymbol{x}_2 = [0, 10]$. Для первого уравнения $y = 4 - x^2$ на интервале $\boldsymbol{x}_2 = [0, 10]$ находим $\boldsymbol{y} = 4 - \boldsymbol{x}^2 = [4, 4] - [0, 10]^2 = [-96, 4]$. Новый $\boldsymbol{y} = [-10, 4] \cap [-96, 4] = [-10, 4]$ (без изменений). Для второго уравнения $y = 5 - e^x$

вычислим $e^x \in \left[e^0, e^{10}\right] \approx [1,\ 22026]$. Тогда $\boldsymbol{y} = 5 - e^x = [5,5] - [1,\ 22026] = [-22021, 4]$. Находим пересечение с исходным интервалом и получим $\boldsymbol{y} = [-10,\ 4] \cap [-22021,\ 4] = [-10,\ 4]$. Дальше нет новых сужений, и требуется дальнейшее разбиение интервала $\boldsymbol{x}_2$.

**2.3.** Разбиваем $\boldsymbol{x}_2 = [0, 10]$ на $\boldsymbol{x}_{21} = [0,\ 5]$ и $\boldsymbol{x}_{22} = [5, 10]$. Для $\boldsymbol{x}_{22} = [5, 10]$: $x^2 \in [25, 100] \rightarrow y \in [-96, -21]$, но из $y = 5 - e^x$, $e^x \in \left[e^5, e^{10}\right] \approx [148,\ 22026]$. Тогда $\boldsymbol{y} = 5 - e^x = [5,5] - [148,\ 22026] = [-22021, -143]$. Находим пересечение с исходным интервалом и получим $\boldsymbol{y} = [-96, -21] \cap [-22021, -143] = \varnothing$, а значит, интервал исключается. Теперь обрабатываем $\boldsymbol{x}_{21} = [0, 5]$: Аналогично находим, что единственное возможное решение: $x = 0, \quad y = 4$.

Из графика для системы (заданное на примере 2) нелинейных уравнений видно (Рис 1.), что система при интервальных ограничениях имеет единственное решение в точке $(0,4)$.

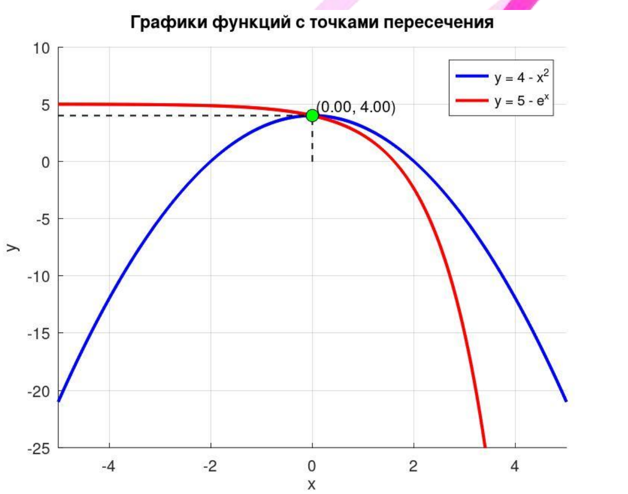


Рис. 1. Графики функций для примера 2.

Интервальные методы распространения ограничений (ICP) является мощным инструментом для решения **нелинейных уравнений** и **оптимизации,** обработки **непрерывных доменов** с гарантированной точностью, а также для работы (в комбинации) с другими методами. Их основное преимущество — это **устойчивость к ошибкам округления** и возможность работы там, где символические методы неэффективны.

Данный подход широко применяется в глобальной оптимизации, робототехнике и верификации моделей. Для сложных систем может потребоваться оптимизация (например, выбор стратегии разбиения).

## Литература

1. Bessiere C. (1994) Using constraint metaknowledge to reduce arc

consistency computation. *Artificial Intelligence* **65**, pp. 179-190.

2. Bessiere C., Freuder E.C., and Régin J.-R. (1999) Comments on Mohr and Henderson's path consistency algorithm. *Artificial Intelligence* 107, pp. 125-148.
3. Benhamou F., McAllester D., Van Hentenryck P. CLP (Intervals) Revisited. In: Proc. of ILPS'94, Ithaca, NY, USA, 1994, pp. 124−138.
4. Benhamou F., Older W. Applying Interval Arithmetic to Real, Integer and Boolean Constraints. Journal of Logic Programming 32 (1997), pp.1-24.
5. Benhamou, F., Granvilliers, L., Goualard, F. (2000) Interval Constraints: Results and Perspectives. In: Apt, K.R., Monfroy, E., Kakas, A.C., Rossi, F. (eds) New Trends in Constraints. WC 1999. Lecture Notes in Computer Science, vol. 1865. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44654-0_1
6. Schichl H., Neumaier A. (2005) Interval Analysis on Directed Acyclic Graphs for Global Optimization. Journal of Global Optimization, 33(4), pp. 541–562.
7. **Chabert G., Jaulin L. (2009)** *Contractor Programming. Artificial Intelligence*, 173(11), pp. 1079–1100.
8. **Goldsztejn A. (2012) Bounds Consistency for Continuous Constraints. *Constraints*, 17(1), 1–24.**
9. **Sotiropoulos, D.G., Grapsa, T.N. (2001) A Branch-and-Prune Method for Global Optimization. In: Krämer, W., von Gudenberg, J.W. (eds) Scientific Computing, Validated Numerics, Interval Methods. Springer, Boston, MA.** https://doi.org/10.1007/978-1-4757-6484-0_18
10. **Ибрагимов А.А., Базаров М.Б., Шокин Ю.И., Юлдашев З.Х. (2013) Математическое моделирование интервальными методами. - Ташкент: Фан, 160 с.**