Российская академия наук
Сибирское отделение
Институт систем информатики
им. А. П. Ершова

М. Ю. Лоенко

УЛУЧШЕНИЕ ВНЕШНЕЙ ОЦЕНКИ МНОЖЕСТВА РЕШЕНИЙ ЗАДАЧ УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ

Препринт
79

Новосибирск 2000

В настоящее время существуют эффективные методы решения нелинейных уравнений с одной переменной, такие как интервальный метод Ньютона, метод Кравчика и другие [3]. Эти методы неприменимы или неэффективны при решении систем уравнений. В статье представлен алгоритм NC, предназначенный для улучшения существующей внешней оценки множества решений. Предлагаемый алгоритм использует методы решения уравнений с одной переменной для решения систем уравнений с несколькими переменными.

Siberian Division of the Russian Academy of Sciences
A. P. Ershov Institute of Informatics Systems

M. Yu. Loenko

IMPROVING OF AN EXTERNAL ESTIMATION OF THE SOLUTION SET TO CONSTRAINT SATISFACTION PROBLEMS

Preprint
79

Novosibirsk 2000

At present, there are effective algorithms for solving non-linear univariate equations, for example, the interval Newton method. These methods are either ineffective or unapplicable to systems of multi-variable equations. The paper presents the NC algorithm for improving an external estimation of the solution set to a constraint satisfaction problem. To solve multi-variable systems, it uses existing methods for solving univariate equations.

1. ВВЕДЕНИЕ

Статья посвящена решению численных задач удовлетворения ограничений. Численные задачи удовлетворения ограничений определяются как тройка (X, D, C), где:

X — множество переменных $\{x_1, ..., x_n\}$,

$D = D_1 \times ... \times D_n$, где D_i — множество значений переменной x_i,

C — множество ограничений $\{c_1, ..., c_m\}$, c_j — отношение (уравнение, неравенство, таблица), связывающее некоторые переменные $x_{j_1}, ..., x_{j_k}$.

Далее под словом *задача* будем понимать численную задачу удовлетворения ограничений. *Решением* задачи $M = (X, D, C)$ называется любой вектор $(a_1, ..., a_n) \in D$ такой, что если $c_i(x_{i_1}, ..., x_{i_k}) \in C$, то $(a_{i_1}, ..., a_{i_k}) \in c_i$.

Необходимость решения таких задач возникает при моделировании физических и химических процессов, в системах автоматического проектирования и т. д. При этом в зависимости от конкретного случая может возникнуть необходимость найти либо все решения задачи, либо любое решение, или как можно меньшее множество, содержащее все решения задачи. Часто в качестве такого множества используют многомерный интервал.

Многомерный замкнутый интервал $I = I_1 \times ... \times I_n$ называется *внешней оценкой* решения задачи M, если он содержит все решения задачи M. Внешняя оценка решения задачи M называется *оптимальной*, если она содержится в любой внешней оценке решения M.

Нахождение оптимальной внешней оценки решения в общем случая является NP-трудной задачей и поэтому не всегда возможно. Следовательно, алгоритмы, позволяющие находить *субоптимальную* внешнюю оценку решения, т. е. какую-нибудь внешнюю оценку, в некотором смысле близкую к оптимальной, имеют практическую ценность. Поскольку исходное множество D уже является внешней оценкой решения, то обычно вопрос стоит не о нахождении, а об улучшении внешней оценки.

В этой статье предлагается алгоритм NC — метод улучшения внешней оценки решения численных задач удовлетворения ограничений. Он основан на построении несовместной подзадачи $M^{'}$ исходной задачи M, набора характеристических уравнений несовместности $M^{'}$ и решении этих уравнений.

Статья организована следующим образом: сначала описан алгоритм M2B — один из наиболее известных алгоритмов построения внешней оценки решения. Алгоритм NC использует некоторые механизмы алгоритма M2B. Затем даны базовые понятия для описания алгоритма NC. Далее представлен сам алгоритм и доказана его корректность. В заключение продемонстрированы результаты сравнений вычислительной сложности алгоритмов NC и M2B и результаты экспериментов.

2. СОГЛАШЕНИЯ

В этом разделе приведены принятые в статье соглашения, обозначения, базовые определения.

Обозначим $\overline{R}$ — множество вещественных чисел, расширенное элементами $-\infty$ и ∞.

Обозначим FP — множество машиннопредставимых чисел, то есть чисел, точно представимых в некотором машинном формате. В этой работе предполагается, что числа $-\infty$ и ∞ являются машиннопредставимыми. Машиннопредставимые числа будем называть также FP-числами. Пусть a — вещественное число, обозначим:

$a^- = sup\ \{x \in FP | x \leq a\}$,

$a^+ = inf\ \{x \in FP | x \geq a\}$.

Заметим, что поскольку множество FP — конечно, то для всех $a \in \overline{R}$ верно: $a^-, a^+ \in FP$.

Будем полагать, что $inf\ \emptyset = \infty$ и $sup\ \emptyset = -\infty$.

Пусть дана задача $M = (X, D, C)$. Здесь и далее мы будем полагать, что все переменные — вещественные, D — многомерный интервал, границы которого — FP-числа, а множество C может содержать только следующие виды отношений:

- $\{(x, y) | x = y\}$,
- $\{(x, y) | x \leq y\}$,
- $\{(x, y) | x = -y\}$,
- $\{(x, y) | x = |y|\}$,
- $\{(x, y) | x = \sin y\}$,
- $\{(x, y) | x = \cos y\}$,
- $\{(x, y) | x = \tan y\}$,
- $\{(x, y) | x = \cot y\}$,
- $\{(x, y) | x = e^y\}$,
- $\{(x, y) | x = y^n, n - \text{ целое}\}$,
- $\{(x, y) | x = y + a\}$,
- $\{(x, y) | x = ay, a \neq 0\}$,
- $\{(x, y, z) | x = y + z\}$,
- $\{(x, y, z) | x = yz\}$.

Определение. Пусть $A \subset \overline{R}^k$, множество $\{x_i | \exists x_1, ..., x_{i-1}, x_{i+1}, ..., x_k \mid (x_1, ..., x_k) \in A\}$ будем называть i-той проекцией множества A и обозначать $P_i(A)$.

Лемма 1. Пусть даны множества $A_2, ..., A_n \subset \overline{R}$ и $B \subset \overline{R}^n$, обозначим $E = P_1(\overline{R} \times A_2 \times ... \times A_n \cap B)$. Тогда для любого $A_1 \subset \overline{R}$ верно: $P_1(A_1 \times A_2 \times ... \times A_n \cap B) = A_1 \cap E$.

Доказательство. Пусть $x_1 \in P_1(A_1 \times A_2 \times ... \times A_n \cap B)$, тогда существуют $x_2, ..., x_n$ такие, что $(x_1, ..., x_n) \in A_1 \times A_2 \times ... \times A_n \cap B$. Тогда $x_1 \in A_1$ и $(x_1, ..., x_n) \in \overline{R} \times A_2 \times ... \times A_n \cap B$, следовательно, $x_1 \in A_1 \cap E$.

Пусть $x_1 \in A_1 \cap E$. Тогда существуют $x_2, ..., x_n$, такие что $(x_1, ..., x_n) \in \overline{R} \times A_2 \times ... \times A_n \cap B$. Поскольку $x_1 \in A_1$, то $(x_1, ..., x_n) \in A_1 \times ... \times A_n \cap B$. Следовательно, $x_1 \in P_1(A_1 \times ... \times A_n \cap B)$. Лемма доказана.

Определение. Задачи $M = (X, D, C)$ и $M' = (X', D', C')$ называются эквивалентными, если $X = X'$ и множества решений задач M и M' совпадают.

Определение. Задача M называется *2B-совместной* [5], если для любого ограничения $c_j(x_{j_1}, ..., x_{j_k}) \in C$, для всех $i = 1, ..., k$, если $A_i = P_i(D_{j_1} \times ... \times D_{j_k} \cap c_j)$, то $D_{j_i} = [(inf\ A_i)^-, (sup\ A_i)^+]$.

3. АЛГОРИТМ M2B

Этот раздел описывает один из существующих в настоящее время алгоритмов улучшения оценки решения — алгоритм M2B.

Пусть дана задача $M = (X, D, C)$. Построим двудольный ориентированный граф G с вершинами двух типов: *вершинами-переменными* и *вершинами-отношениями* — так, чтобы каждой переменной $x_i \in X$ соответствовала некоторая вершина-переменная, каждому k-арному отношению $c_j(x_{j_1}, ..., x_{j_k}) \in C$ соответствовали k вершин-отношений, имеющих по одной выходящей дуге и по k входящих. Причем, если вершины-переменные $v_1, ..., v_k$ соответствуют переменным $x_{j_1}, ..., x_{j_k}$, то для любого i существует вершина-отношение r такая, что граф G содержит дуги: (v_1, r), ..., (v_k, r) и (r, v_i).

Граф G будем называть *графом задачи*, соответствующим задаче M.

Если вершина-отношение r имеет выходящую дугу (r, v), то вершину v будем обозначать $t(r)$.

Каждой вершине-переменной поставлено в соответствие ее *значение*. Значением вершины-переменной является интервал $[a, b]$, где $a, b \in FP$, при этом, если $a > b$, будем говорить, что значение пусто. При построении графа значения вершин-переменных устанавливаются равными областям значений соответствующих переменных в задаче M. Будем говорить, что граф *пустой*, если существует вершина-переменная, значение которой пусто, в противном случае — граф *непустой*.

Каждая вершина-отношение может быть *активна*. Будем говорить, что граф задачи *активный*, если в нем существует хотя бы одна активная вершина-отношение.

Каждая вершина-отношение может быть *исполнена*. Пусть $c_j(x_{j_1}, ..., x_{j_k}) \in C$, вершина-отношение r соответствует ограничению c_j, вершины $v_1, ..., v_k$ соответствуют переменным $x_{j_1}, ..., x_{j_k}$ задачи M, $t(r) = v_m$, где $1 \leq m \leq k$. Пусть $I_1, ..., I_k$ — значения вершин $v_1, ..., v_k$. Тогда исполнение вершины r — это вычисление значения $I_m^{'} := [(inf\ A)^-,\ (sup\ A)^+]$, где $A = P_m(c_j \cap I_1 \times ... \times I_k)$, и установление $I_m^{'}$ в качестве значения вершины v_m.

Лемма 2. Пусть вершина-отношение r графа G соответствует ограничению $c_j(x_{j_1}, ..., x_{j_k}) \in C$. Пусть вершины $v_1, ..., v_k$ соответствуют переменным $x_{j_1}, ..., x_{j_k}$. Пусть $I_1, ..., I_k$ — значения вершин $v_1, ..., v_k$. Пусть исполнение r меняет значение вершины v_m, где $1 \leq m \leq k$. Пусть $I_m^{'}$ — значение v_m после исполнения r. Пусть $inf\ I_m, sup\ I_m \in FP$. Тогда $I_m^{'} \subset I_m$ и $inf\ I_m^{'}, sup\ I_m^{'} \in FP$.

Доказательство. Пусть $A = P_m(c_j \cap I_1 \times ... \times I_k)$. Тогда $I_m^{'} = [(inf\ A)^-,\ (sup\ A)^+]$. Поскольку для любого $a \in \overline{R}$ верно: $a^-, a^+ \in FP$, то $(inf\ A)^-, (sup\ A)^+ \in FP$, а значит и $inf\ I_m^{'}, sup\ I_m^{'} \in FP$.

Далее из определения проекции следует, что $A \subset I_m$, поэтому $inf\ A \geq inf\ I_m$, следовательно, $sup\ \{x \in FP | x \leq inf\ A\} \geq sup\ \{x \in FP | x \leq inf\ I_m\}$, что эквивалентно $(inf\ A)^- \geq (inf\ I_m)^-$. Поскольку $inf\ I_m \in FP$, то $(inf\ I_m)^- = inf\ I_m$. Значит, $(inf\ A)^- \geq inf\ I_m$.

Аналогично доказывается, что $(sup\ A)^+ \leq sup\ I_m$. Поскольку I_m — интервал, то $I_m^{'} \subset I_m$. Лемма доказана.

Лемма 3. Пусть $v_1, ..., v_n$ — все вершины-переменные графа G. Пусть они соответствуют переменным $x_1, ..., x_n \in X$, пусть $I_1, ..., I_n$ — значения вершин $v_1, ..., v_n$. Пусть вершина-отношение r соответствует ограничению $c_j(x_{j_1}, ..., x_{j_k}) \in C$. Пусть исполнение вершины r меняет значение вершины v_{j_m}. Пусть $I_{j_m}^{'}$ — значение вершины v_{j_m} после исполнения r. Обозначим $I = I_1 \times ... \times I_n$, $I^{'} = I_1 \times ... \times I_{j_m - 1} \times I_{j_m}^{'} \times I_{j_m + 1} ... \times I_n$. Тогда задачи $M_1 = (X, I, C)$ и $M_2 = (X, I^{'}, C)$ эквивалентны.

Доказательство. Поскольку множества переменных задач совпадают, то для доказательства эквивалентности достаточно доказать совпадение множеств решений задач M_1 и M_2. Из леммы 2 следует, что $I^{'} \subset I$, следовательно, все решения задачи M_2 являются решениями задачи M_1.

Предположим, что $a \in I$ — решение задачи M_1. Тогда $(a_{j_1}, ..., a_{j_k}) \in I_{j_1} \times ... \times I_{j_k}$ и $(a_{j_1}, ..., a_{j_k}) \in c_j$, следовательно, $a_{j_m} \in P_m(c_j \cap I_{j_1} \times ... \times I_{j_k})$. А по определению исполнения вершины-отношения $a_{j_m} \in I^{'}_{j_m}$. Следовательно, $a \in I^{'}$ и является решением задачи M_2. Лемма доказана.

Если G — активный непустой граф задачи, то *итерацией алгоритма M2B* будем называть выполнение следующих действий:

1) выбор согласно некоторому правилу активной вершины-отношения r;

2) исполнение вершины r;

3) активизация всех вершин-отношений $r^{'}$ таких, что граф G содержит дугу $(t(r), r^{'})$; если при исполнении отношения изменилось значение вершины-переменной $t(r)$;

4) деактивизация вершины r.

Определение. Пусть G — граф задачи, $v_1, ..., v_n$ — его вершины-переменные, $I_1, ..., I_n$ — их значения. Тогда число $(\sum |I_i \cap FP|)N_r + A$, где $|X|$ — число элементов множества X, N_r — количество вершин-отношений, а A — количество активных вершин-отношений, будем называть *магическим* числом графа G. Магическое число любого графа задачи конечно, это следует из конечности множества FP. Магическое число любого графа задачи неотрицательно.

Лемма 4. Итерация алгоритма M2B уменьшает магическое число графа задачи.

Доказательство. Пусть при итерации алгоритма была исполнена некоторая вершина r. Предположим, что в результате значение вершины $t(r)$ изменилось, тогда по определению исполнения вершины новое значение содержится в старом. Следовательно, число $\sum |I_i \cap FP|$ уменьшилось. Поскольку до исполнения отношения количество активных вершин было не меньше единицы, а после стало не больше $N_r - 1$, то магическое число уменьшилось. Если в результате итерации значение вершины $t(r)$ не изменилось, то сумма $\sum |I_i \cap FP|$ осталась прежней, а количество активных вершин уменьшилось на единицу, а значит, магическое число также уменьшилось. Лемма доказана.

Алгоритм M2B сформулирован следующим образом:

1) по задаче M строится граф задачи G;

2) все вершины-отношения становятся активными;

3) пока граф активен и не пуст, последовательно производятся итерации алгоритма M2B;

4) в случае, если граф стал пустым, задача M несовместна. В противном случае, если $I_1^{'}, ..., I_n^{'}$ — значения вершин переменных, соответствующих переменным $x_1, ..., x_n$ задачи M, то задача $M^{'} = (X, I_1^{'} \times ... \times I_n^{'}, C)$ эквивалентна исходной задаче M и является 2B-совместной.

Сходимость алгоритма M2B следует из леммы 4, эквивалентность задач M и $M^{'}$ — из Леммы 3. Осталось доказать, что задача $M^{'}$ 2B-совместна.

Доказательство. Предположим, что $M^{'}$ не является 2B-совместной. Это означает, что существуют ограничение $c_j(x_{j_1}, ..., x_{j_k})$ и переменная $x_{j_m} \in X$, где $1 \leq m \leq k$, такие, что если $A = P_m(I_{j_1}^{'} \times ... \times I_{j_k}^{'} \cap c_j)$, то $I_{j_m}^{'} \neq [(inf\ A)^-, (sup\ A)^+]$.

Пусть вершины $v_1, ..., v_k$ соответствуют переменным $x_{j_1}, ..., x_{j_k}$, а вершина r — отношению c_j, причем $t(r) = v_m$.

Рассмотрим последнюю итерацию алгоритма M2B, на которой исполнялась вершина r. Пусть $I_1, ..., I_k$ — значения вершин $v_1, ..., v_k$ перед последним исполнением r. Поскольку из всех вершин $v_1, ..., v_k$ идут дуги в вершину r, то значения вершин $v_1, ..., v_k$ не менялись после последнего исполнения r, иначе вершина r исполнилась бы еще раз. Следовательно, для всех $i \in \{1, ..., k\} \backslash \{m\}$ выполняется: $I_i = I_{j_i}^{'}$.

Обозначим $E = P_m(c_j \cap I_1 \times ... \times I_{m-1} \times \overline{R} \times I_{m+1} \times ... \times I_k)$, $B = P_m(c_j \cap I_1 \times ... \times I_k)$. Тогда по лемме 1 $E \cap I_m = B$, $E \cap I_{j_m}^{'} = A$.

Предположим, что $I_{j_m}^{'} \backslash [(inf\ A)^-, (sup\ A)^+] \neq \emptyset$. По определению исполнения вершины-ограничения $I_{j_m}^{'} = [(inf\ B)^-, (sup\ B)^+]$. Пусть $(inf\ B)^- < (inf\ A)^-$. Тогда, поскольку $(inf\ A)^- \in FP$, то $inf\ B < (inf\ A)^-$. Следовательно, существует $x \in B$ такое, что $x < (inf\ A)^-$. Поскольку $x \in B$, то $x \in E$ и $x \in [(inf\ B)^-, (sup\ B)^+] = I_{j_l}^{'}$. Следовательно, $x \in E \cap I_{j_l}^{'} = A$. Но $x < (inf\ A)^-$, поэтому приходим к противоречию. Следовательно, $(inf\ A)^- \leq (inf\ B)^-$. Аналогично доказывается, что $(sup\ B)^+ \leq (sup\ A)^+$. Значит, $I_{j_m}^{'} \subset [(inf\ A)^-, (sup\ A)^+]$.

Из леммы 2 следует, что $[(inf\ A)^-, (sup\ A)^+] \subset I_{j_m}^{'}$, поэтому $I_{j_m}^{'} = [(inf\ A)^-, (sup\ A)^+]$. Получается противоречие. Задача $M^{'}$ 2B-совместна.

4. ФУНКЦИИ-ХАРАКТЕРИСТИКИ И УСЛОВИЯ КОРРЕКТНОСТИ

Пусть дана задача $M = (X, D, C)$, граф задачи G соответствует задаче M, r — вершина-отношение графа G, $c_l(x_{l_1}, ..., x_{l_u})$ — соответствующее ей отношение в C. Пусть вершины-переменные $v_{l_1}, ..., v_{l_u}$ графа G соответствуют переменным $x_{l_1}, ..., x_{l_u}$ задачи M, при этом $t(r) = v_{l_k}$, $1 \leq k \leq u$. $I_{l_1}, ..., I_{l_u}$ — значения вершин $v_{l_1}, ..., v_{l_u}$. Обозначим

$A = P_m(c_j \cap I_{l_1} \times ... \times I_{l_u})$,

$y = (inf\ I_{l_1}, sup\ I_{l_1}, ..., inf\ I_{l_u}, sup\ I_{l_u}, p_1, ..., p_s)$,

тогда тогда получаем следующие определения.

Нижней характеристикой исполнения r с вектором параметров $(p_1, ..., p_s) \in FP^s$, где $s \geq 0$, называется любая функция $F_0 : \overline{R}^{2u+s} \to \overline{R}$ такая, что $F_0(y) \leq inf\ A$.

При этом *набором условий корректности* нижней характеристики F_0 с вектором параметров $(p_1, ..., p_s)$ называется множество функций

$F_1 : \overline{R}^{2u+s} \to \overline{R}$,

...

$F_t : \overline{R}^{2u+s} \to \overline{R}$,

таких, что выполняется:

1) $\forall i\ F_i(y) \geq 0$
2) если $J \subset \overline{R}^u$ — u-мерный интервал, $y_1 = (inf\ J_{l_1}, sup\ J_{l_1}, ..., inf\ J_{l_u}, sup\ J_{l_u}, p_1, ..., p_s)$, и для всех i таких, что $1 \leq i \leq t$, верно $F_i(y_1) > 0$, то $F_0(y_1) \leq inf\ A'$, где $A' = P_k(c_l \cap J_{l_1} \times ... \times J_{l_u})$.

При этом функции $F_1, ..., F_t$ называются условиями корректности F_0.

Неформальным языком нижнюю характеристику можно определить как "ту самую функцию", которая используется для вычисления нижней границы интервала при исполнении вершины-отношения. При этом набор условий корректности — это набор условий, при выполнении которых используется именно "та" функция. С помощью вектора параметров можно достигнуть уменьшения необходимого набора таких функций, например, функции $F_1(x) := x + 1$ и $F_2(x) := x + 2$ могут быть представлены одной функцией $F(x, y) := x + y$ с вектором параметров, состоящим из одного элемента.

Аналогично определяются *верхняя характеристика* исполнения вершины-отношения и набор условий ее корректности.

Теорема 1. Существует конечный набор $\Phi = \{F_i : \overline{R}^{m_i} \to \overline{R}\}$ функций, являющихся конечными суперпозициями элементарных функций, такой, что для любой задачи M, если G — граф задачи, соответствующий M, r — вершина-отношение графа G, то существуют функции $F_{l_0}, ..., F_{l_t}, F_{u_0}, ..., F_{u_s} \in \Phi$, такие что F_{l_0} — нижняя характеристика исполнения r; $F_{l_1}, ..., F_{l_t}$ — набор условий корректности F_{l_0} и F_{u_0} — верхняя характеристика исполнения r; $F_{u_1}, ..., F_{u_s}$ — набор условий корректности F_{u_0}. Набор Φ будем называть *набором исторических функций.*

Доказательство. Рассмотрим набор функций:
$F_1(x_1, ..., x_4) = -\infty$,
$F_2(x_1, ..., x_4) = \infty$,
$F_3(x_1, ..., x_6) = -\infty$,
$F_4(x_1, ..., x_6) = \infty$.
Если вершина-отношение r соответствует бинарному ограничению $c_j(x_{j_1}, x_{j_2})$, то функция F_1 будет нижней, а функция F_2 — верхней характеристикой исполнения r. Это следует из определения характеристик исполнения. Аналогично, если вершина-отношение r соответствует 3-арному ограничению $c_l(x_{l_1}, x_{l_2}, x_{l_3})$, то функция F_3 будет нижней, а функция F_4 — верхней характеристикой исполнения r. Во всех четырех случаях векторы параметров и наборы условий корректности будут пустыми. Поскольку все отношения задачи M — бинарные или 3-арные, теорема доказана.

Заметим, что набор функций, приведенных в доказательстве, является абсолютно бесполезным на практике. В реальности можно использовать функции, применяемые для вычислений границ элементарных математических функций и операций над интервалами [4].

Определение. Пусть $F : \overline{R}^k \to \overline{R}$ — историческая функция. Функцию $\widehat{F} : \overline{R}^w \to \overline{R}$, где $w \leq k$, которая удовлетворяет требованиям:

1. $\exists e_1, ..., e_w | \forall (x_1, ..., x_k)\ \widehat{F}(x_{e_1}, ..., x_{e_w}) = F(x_1, ..., x_k)$;
2. $\forall j \in \{1, ..., w\}\ \exists x_{e_1}, ..., x_{e_w}\ \exists x'_{e_j} \mid \widehat{F}(x_{e_1}, ..., x_{e_w}) \neq \widehat{F}(x_{e_1}, ..., x_{e_{j-1}}, x'_{e_j}, x_{e_{j+1}}, ..., x_{e_w})$,

будем называть *упрощением* функции F. При этом функцию F будем называть *усложнением* функции $\widehat{F}$.

5. ГРАФ ИСТОРИИ

Одним из ключевых понятий в описании предлагаемого в статье алгоритма NC является понятие *графа истории.* В графе истории сохраняется информация о работе алгоритма M2B. В этом разделе будут описаны типы вершин графа истории и его возможные преобразования.

Граф истории является ориентированным графом с тремя основными типами вершин: вершинами-переменными, вершинами-числами и вершинами-функциями. Он построен на основе некоторого непустого графа задачи, который мы будем называть *базовым* для данного графа истории. С графом истории связан некоторый набор исторических функций.

Как будет описано далее, алгоритм NC сначала создает заготовку графа истории, которая впоследствии может быть изменена в результате *исполнения* некоторой вершины *с сохранением следа в графе истории*. Такие преобразования будут выполняться последовательно, одна итерация алгоритма NC будет производить одно такое преобразование.

Если a — вершина графа истории, то $time(a)$ — номер итерации алгоритма NC, на котором была создана вершина a, либо 0, если она присутствовала в заготовке.

Будем говорить, что вершина a *старше* вершины b, если $time(a) > time(b)$. При этом будем говорить что вершина b *моложе* вершины a. Если $time(a) = time(b)$, будем говорить, что вершины a и b — *ровесники*.

5.1. Вершины графа истории

5.1.1. Вершины-числа

Вершины-числа делятся на *константы* и *результаты*. Вершины-константы в свою очередь делятся на *границы* и *параметры*. Среди вершин-границ выделим пару вершин, которые будем называть *аргументами*. Среди вершин-результатов будем выделять *цели*.

Каждой вершине-константе соответствует ее *значение*. Тип значения вершины-константы — FP-число. Значение вершины-константы будет оговариваться при ее создании, в противном случае оно будет считаться неопределенным (или произвольным).

Вершины-числа могут иметь как входящие, так и выходящие дуги.

5.1.2. Вершины-переменные

Множество вершин-переменных графа истории совпадает с множеством вершин-переменных базового графа задачи. Одна из вершин будет называться *главной*.

Вершины-переменные не имеют входящих дуг. Каждая вершина-переменная имеет две выходящие дуги, каждая из которых идет в некоторую вершину-границу. Выходящие дуги каждой вершины-переменной упорядочены.

Если v — вершина-переменная, то вершину-границу c, такую что (v, c) — первая выходящая дуга вершины v, будем обозначать $l(v)$.

Если v — вершина-переменная, то вершину-границу c, такую что (v, c) — вторая выходящая дуга вершины v, будем обозначать $h(v)$.

По ходу пополнения графа истории некоторые дуги вида (v, c), где v — вершина-переменная, c — вершина-граница, будут удаляться и вместо них создаваться новые. Однако для доказательства корректности графа алгоритма NC нам потребуется ссылаться на вершины-границы, в которые шли дуги из определенных вершин-переменных на определенной итерации алгоритма NC. В этом случае, если v — вершина-переменная, то вершину-границу c, такую что после k итераций алгоритма NC дуга (v, c) стала первой выходящей дугой вершины v, будем обозначать $l_k(v)$.

Аналогично, если v — вершина-переменная, то вершину-границу c, такую что после k итераций алгоритма NC дуга (v, c) стала второй выходящей дугой вершины v, будем обозначать $h_k(v)$.

5.1.3. Вершины-функции

Каждая вершина-функция будет иметь связанную с ней функцию *исполнения.* Количество аргументов функции исполнения, связанной с данной вершиной-функцией, совпадает с количеством входящих в нее дуг. Входящие дуги каждой вершины-функции упорядочены. Если f — вершина-функция, вершину-число c, такую что дуга (c, f) существует и является i-той входящей дугой вершины f, будем обозначать $s_i(f)$. Если f — вершина-функция, вершину-число c, такую что существует дуга (f, c), будем обозначать $t(f)$. Вершины-функции делятся на вершины-условия, н-вершины и в-вершины.

5.2. Создание заготовки графа истории

Пусть дан граф непустой задачи G, пусть $v_1, ..., v_n$ — все его вершины-переменные, $I_1, ..., I_n$ — их значения. Пусть некоторая переменная x_i названа *главной* (см. раздел 7.1), v_i — соответствующая x_i вершина-переменная графа G. Пусть $inf\ I_i = sup\ I_i$.

Тогда граф H, который содержит:

1) вершины-переменные $v_1, ..., v_n$ графа G;
2) вершины-границы $c_1, ..., c_{2n}$ со значениями $inf\ I_1, sup\ I_1, ..., inf\ I_n$, $sup\ I_n$, среди которых вершины c_{2i-1} и c_{2i} являются аргументами;
3) дуги (v_k, c_{2k-1}), (v_k, c_{2k}), где $k = 1, ..., n$, при этом (v_k, c_{2k-1}) — первая, а (v_k, c_{2k}) — вторая выходящие дуги вершины v_k;

называется заготовкой графа истории, построенной на базе графа задачи G. Заготовка графа истории также является графом истории. Заготовка графа истории не имеет вершин-функций.

5.3. Эволюция графа истории

После создания графа истории происходит его эволюция. Эволюция графа истории будет происходить по мере работы алгоритма NC. Изменения графа истории возможны только в рамках *исполнения ограничений с сохранением следа в графе истории.* Для описания этого преобразования введем следующие промежуточные преобразования:

- создание следа границы;
- сохранение характеристики исполнения;
- сохранение условий корректности вершины-функции.

Опишем все преобразования подробно.

5.3.1. Создание следа границы

Пусть H — граф истории, r — главная вершина некоторой итерации алгоритма NC, $v = t(r)$ — вершина-переменная, c — вершина-число.

Преобразование *создание следа нижней границы* заключается в замене первой выходящей дуги вершины v дугой (v, c). Это преобразование будем обозначать $\underline{trace}(H, v, c)$.

Преобразование *создание следа верхней границы* заключается в замене второй выходящей дуги вершины v дугой (v, c). Его будем обозначать $\overline{trace}(H, v, c)$.

5.3.2. Сохранение характеристики исполнения

Пусть r — вершина-отношение графа задачи G соответствует ограничению $c_l(x_{l_1}, ..., x_{l_u})$, вершины $v_{l_1}, ..., v_{l_u}$ соответствуют переменным $x_{l_1}, ..., x_{l_u}$.

Пусть при исполнении r меняется нижняя граница значения вершины $t(r)$. Пусть функция $F : \overline{R}^{2u+s} \to \overline{R}$ — нижняя характеристика исполнения r с вектором параметров $(p_1, ..., p_s)$, функция $\widehat{F} : \overline{R}^w \to \overline{R}$ — упрощение F, и $F(y_1, ..., y_{2u+s}) = \widehat{F}(y_{e_1}, ..., y_{e_w})$.

Тогда *сохранение нижней характеристики* исполнения r — это выполнение следующих действий:

1) создание s вершин-параметров $c_1, ..., c_s$ со значениями $p_1, ..., p_s$;
2) создание вершины-результата c_0;
3) создание н-вершины f со связанной с ней функцией исполнения $\widehat{F}$;
4) создание дуги (f, c_0) и дуг по порядку: (b_k, f), где $k = 1, ..., w$,

$$b_k = \begin{cases} c_{e_k-2u}, & \text{если } e_k > 2u; \\ h(v_{\frac{e_k+1}{2}}), & \text{если } e_k \le 2u,\ e_k \text{ — четное;} \\ l(v_{\frac{e_k}{2}}), & \text{если } e_k \le 2u,\ e_k \text{ — нечетное.} \end{cases}$$

Будем говорить, что н-вершина f *характеризует* исполнение r *снизу*. Вершины $c_1, ..., c_s$ будем называть *параметрами* вершины f.

Если при исполнении r меняется верхняя граница значения вершины $t(r)$, функция $F : \overline{R}^{2u+s} \to \overline{R}$ — верхняя характеристика исполнения r с вектором параметров $(p_1, ..., p_s)$, функция $\widehat{F} : \overline{R}^w \to \overline{R}$ — упрощение F, и $F(y_1, ..., y_{2u+s}) = \widehat{F}(y_{e_1}, ..., y_{e_w})$.

Тогда *сохранение верхней характеристики* исполнения r — это выполнение следующих действий:

1) создание s вершин-параметров $c_1, ..., c_s$ со значениями $p_1, ..., p_s$;
2) создание вершины-результата c_0;
3) создание в-вершины f со связанной с ней функцией исполнения $\widehat{F}$;
4) создание дуги (f, c_0) и дуг по порядку: (b_k, f), где $k = 1, ..., w$,

$$b_k = \begin{cases} c_{e_k-2u}, & \text{если } e_k > 2u; \\ h(v_{\frac{e_k+1}{2}}), & \text{если } e_k \le 2u,\ e_k \text{ — четное;} \\ l(v_{\frac{e_k}{2}}), & \text{если } e_k \le 2u,\ e_k \text{ — нечетное.} \end{cases}$$

Будем говорить, что в-вершина f *характеризует* исполнение r *сверху*.

5.3.3. Сохранение условий корректности

Пусть r — вершина-отношение графа задачи G соответствует ограничению $c_l(x_{l_1}, ..., x_{l_u})$, вершины $v_{l_1}, ..., v_{l_u}$ соответствуют переменным

$x_{l_1}, ..., x_{l_u}$. Пусть вершина f графа истории характеризует исполнение r, $(p_1, ..., p_s)$ — вектор параметров f, вершины-параметры $c_1, ..., c_s$ соответствуют параметрам $p_1, ..., p_s$. Пусть $F : \overline{R}^{2u+s} \to \overline{R}$ — усложнение связанной с вершиной f функции, F_i — условие корректности F, $\widehat{F}_i : \overline{R}^{w} \to \overline{R}$ — упрощение функции F_i, и $F_i(y_1, ..., y_{2u+s}) = \widehat{F}_i(y_{e_1}, ..., y_{e_w})$.

Тогда *сохранение условия корректности* F_i в графе истории это:

1) создание вершины-цели t_i;
2) создание вершины-условия f_i со связанной функцией исполнения $\widehat{F}_i$;
3) создание дуг (f_i, t_i), $(t_i, t(f))$ и дуг по порядку: (b_k, f_i), где $k = 1, ..., w$,

$$b_k = \begin{cases} c_{e_k - 2u}, & \text{если } e_k > 2u; \\ h(v_{\frac{e_k+1}{2}}), & \text{если } e_k \leq 2u,\ e_k \text{ — четное;} \\ l(v_{\frac{e_k}{2}}), & \text{если } e_k \leq 2u,\ e_k \text{ — нечетное.} \end{cases}$$

Если F_1, ..., F_t — некоторый набор условий корректности F, то *сохранением набора условий корректности* в графе истории будем называть последовательное сохранение всех условий корректности F_1, ..., F_t. При этом множество созданных вершин-условий $f_1, ..., f_t$ будем называть *набором условий корректности вершины* f.

5.3.4. Исполнение с сохранением

Пусть r — активная вершина-отношение графа задачи G, $v = t(r)$, тогда *исполнение вершины r с сохранением следа в графе истории* это исполнение вершины r в графе задачи G и, в случае изменения значения вершины v, сохранение информации об исполнении вершины r в графе истории H.

То есть, если в результате исполнения изменилась нижняя граница значения вершины v, то при условии, что в результате сохранения нижней характеристики исполнения r не появляется путь из вершины $l(v)$ в создаваемую при сохранении н-вершину, производится сохранение нижней характеристики исполнения r и следа нижней границы v: $\underline{trace}(H, v, t(f))$, где f — созданная при сохранении нижней характеристики н-вершина. В том случае, если сохранение приводит к появлению указанного пути, оно не производится, а нижняя граница значения вершины v восстанавливается в то состояние, в котором она была до исполнения r.

Аналогично, независимо от того, изменилась ли нижняя граница, если изменилась верхняя граница значения вершины v, то при условии, что в результате сохранения верхней характеристики исполнения r не появляется путь из вершины $h(v)$ в создаваемую при сохранении в-вершину, производится сохранение верхней характеристики исполнения r и следа верхней границы v: $\overline{trace}(H, v, t(g))$, где g — созданная при сохранении верхней характеристики в-вершина. Аналогично, если сохранение приводит к появлению указанного пути, оно не производится, а верхняя граница значения вершины v восстанавливается в то состояние, в котором она была до исполнения r.

На практике для проверки упомянутого выше условия можно сохранить соответствующую характеристику исполнения r и, если появится указанный выше путь, удалить все вершины и дуги, созданные в результате этого сохранения. Однако для доказательства корректности NC алгоритма удобнее считать, что эти вершины и дуги просто не создаются. Такое ограничение на исполнения отношений сделано для исключения так называемых циклических явлений [6] алгоритма M2B.

5.4. Свойства графа истории

Лемма 5. Если f — вершина-функция, то вершины f и $t(f)$ — ровесники. Если $s_i(f)$ — вершина-параметр, то f и $s_i(f)$ — также ровесники, иначе $s_i(f)$ старше f.

Доказательство. По определению эволюции графа истории вершина-функция f была создана либо при сохранении характеристики некоторого исполнения, либо при сохранении условий корректности некоторой вершины-функции.

По определению сохранения характеристики исполнения и по определению сохранения условий корректности в обоих случаях одновременно создается некоторая вершина-цель t и дуга (f, t), причем вершины t и f — ровесники. Поскольку входящие и исходящие дуги вершин-функций, будучи единожды созданы, впоследствии не могут быть удалены из графа истории, то $t(f) = t$.

По определению сохранения характеристики исполнения и сохранения условий корректности и, поскольку входящие и исходящие дуги вершин-функций, будучи единожды созданы, не могут быть удалены из графа истории, то верно, что если $s_i(f)$ — вершина-параметр, то f и $s_i(f)$ — ровесники, иначе, $s_i(f)$ старше f.

Лемма 6. Пусть r — вершина-отношение, обозначим $v = t(r)$. Пусть н-вершина f характеризует исполнение r снизу. Пусть $i = time(f)$. Тогда если $j \geq i$, то $time(l_j(v)) \geq i$.

Аналогично, если в-вершина g характеризует исполнение r сверху, $i = time(g)$, и, если $j \geq i$, то $time(h_j(v)) \geq i$.

Доказательство. Докажем первую часть леммы. Предположим, что $i_1 = time(l_j(v)) < i$. Пусть $c = l_j(v)$. Тогда дуга (v, c) была создана на итерации i_1 как первая выходящая дуга вершины v. Однако после i итераций алгоритма NC первой выходящей дугой вершины v была дуга $(v, l_i(v))$. Следовательно, дуга (v, c) была удалена из графа истории на некоторой итерации с номером, не превосходящим i. Поэтому равенство $c = l_j(v)$ невозможно. Следовательно, $time(l_j(v)) \geq i$. Первая часть леммы доказана.

Вторая часть леммы для случая с в-вершиной доказывается аналогично.

6. ГРАФ ИНДИКАТОР НЕСОВМЕСТНОСТИ

Еще одно ключевое понятие алгоритма NC — понятие *графа-индикатора несовместности.* Граф-индикатор несовместности строится на базе пустого графа истории и некоторого интервала, называемого *стартовым.* Границы стартового интервала — FP-числа.

Пусть H — пустой граф истории, пусть значение вершины-переменной графа v пусто. Создадим вершину-цель c_0, вершину-условие f со связанной функцией $F(x_1, x_2) := x_1 - x_2$ и дуги $(f, c_0), (l(v), f), (h(v), f)$ так, чтобы дуга $(l(v), f)$ была первой входящей дугой вершины f.

Граф-индикатор несовместности IN, построенный на базе H, — это граф, который содержит:

1) все вершины-числа и вершины-функции графа H, из которых достижима хотя бы одна из вершин $l(v)$ или $h(v)$;

2) сами вершины $l(v)$ и $h(v)$;

3) обе вершины-аргументы, независимо от того, достижима ли из них одна из вершин $l(v)$ или $h(v)$;

4) все дуги графа H, которые связывают вершины, включенные в граф IN, за исключением дуг вида (c_1, c_2), где c_1, c_2 — вершины числа;

5) вершину-цель c_0, вершину-условие f, дуги $(f, c_0), (l(v), f), (h(v), f)$, причем дуга $(l(v), f)$ — первая входящая дуга вершины f.

Будем считать вершины f и c_0 моложе всех остальных. Вершину c_0 будем называть *главной целью* графа IN.

Далее приведено описание вершин графа IN и описания действий, связанных с графом.

6.1. Вершины-числа

Типы вершин-чисел совпадают с типами вершин-чисел графа истории. Вершина-число может быть *определена* и *не определена.* Изначально все вершины-результаты не определены, а все вершины-константы — определены.

Каждой вершине-числу соответствуют три значения типа интервал, называемые *точное значение*, *интервальное значение* и *интервальная производная.* Границы этих интервалов — FP-числа. Если a — значение вершины-числа c в графе H и эта вершина входит в граф IN, то ее точное и интервальное значения в графе IN будут интервалы $[a, a]$, значение ее интервальной производной — интервал $[0, 0]$, если это не вершина-аргумент, и $[1, 1]$, если это вершина-аргумент. По определению заготовки графа истории изначально соответствующие значения вершин-аргументов графа индикатора попарно одинаковы.

С каждой вершиной-целью t_i графа IN связан набор поисковых интервалов $S_i = \{I_{i_1}, ..., I_{i_k}\}$, границы которых — FP-числа. Изначально для каждой вершины-цели набор ее поисковых интервалов содержит единственный элемент — стартовый интервал. Интервал $I \in S_i$ будем называть *хорошим* поисковым интервалом набора S_i, если $inf\ I = inf\ \{inf\ J|J \in S_i\}$. Хороший интервал $I \in S_i$ будем называть *лучшим* поисковым интервалом набора S_i, если $sup\ I = inf\ \{sup\ J|J \in S_i\ \&\ J$ — хороший$\}$.

Пусть t_i — вершина-цель, $S_i = \{I_{i_1}, ..., I_{i_k}\}$ — набор поисковых интервалов, c_0 — любая вершина-аргумент, x — точное значение, а X — интервальное значение c_0, пусть $X \in S_i$. Тогда *обработкой* вершины t_i будем называть изменение множества его поисковых интервалов $S_i := S_i^{new}$, где:

1) если t_i не определена, то

$$S_i^{new} = (S_i \backslash \{X\}) \cup \{X \cap [inf\ X, inf\ x],\ X \cap [inf\ x, sup\ X]\};$$

2) иначе, если t_i определена, пусть y — точное ее значение, а Y' — интервальная производная, то:

а) если $0 \notin Y$, то

$$S_i^{new} = (S_i \backslash \{X\});$$

б) если $0 \in Y$ и $0 \notin Y'$, то

$$S_i^{new} = (S_i \backslash \{X\}) \ \cup \ \{X \cap [(inf\ X_1)^-, (sup\ X_1)^+]\},$$

где
$X_1 = x - y/Y'$;
в) если $0 \in Y$ и $0 \in Y'$, то
$S_i^{new} = (S_i \backslash \{X\}) \ \cup \ \{X \cap [(inf\ X_1)^-, (sup\ X_1)^+],\ X \cap [(inf\ X_2)^-, (sup\ X_2)^+]\}$,
где
$X_1 = x - ([0, \infty] \cap y/Y')$,
$X_2 = x - ([-\infty, 0] \cap y/Y')$.

При этом обработку вершины-цели будем называть *правильной снизу*, если X — лучший поисковый интервал набора S_i и верно:

1) $x \subset X$;

2) $inf\ X \neq sup\ X \Rightarrow x > inf\ X$.

Лемма 7. Пусть J — объединение поисковых интервалов некоторой вершины-цели t до ее обработки, J^{new} — объединение поисковых интервалов вершины t после обработки, тогда $J^{new} \subset J$.

Доказательство. Поскольку все интервалы, добавляемые в набор поисковых интервалов в результате обработки вершины-цели, получены в результате пересечения некоторого интервала с поисковым интервалом $X \subset J$, то добавляемые интервалы также будут содержаться в J, следовательно $J^{new} \subset J$.

Лемма 8. Пусть t_i — вершина-цель, $S_i = \{I_{i_1}, ..., I_{i_k}\}$ — набор поисковых интервалов, c_0 — любая вершина-аргумент, x — точное значение, а X — интервальное значение c_0, пусть $X \in S_i$. Обозначим J — лучший поисковый интервал набора S_i, а J^{new} — лучший поисковый интервал набора S_i^{new}. Пусть $inf\ J = inf\ J^{new}$. Тогда если обработка t_i — правильная снизу, то либо $J = J^{new}$, либо суммарное количество FP-чисел в хороших интервалах множества S_i больше суммарного количества FP-чисел в хороших интервалах множества S_i^{new}. (Имеется ввиду не количество FP-чисел в объединении хороших интервалов, а сумма количеств FP-чисел по каждому хорошему интервалу).

Доказательство. Заметим, что поскольку обработка t_i — правильная снизу, то $J = X$. Предположим, что вершина t_i не определена, поскольку $x \subset X$, то $J^{new} = [inf\ X, inf\ x]$. Если $inf\ x = sup\ X$, то $J = J^{new}$, что и требовалось доказать. Предположим, что $inf\ x < sup\ X$, поскольку обработка — правильная снизу, то $inf\ X < inf\ x$. Тогда $J^{new} = [inf\ X, inf\ x]$, а интервал $[inf\ x, sup\ X]$ не является хорошим. Поскольку $sup\ J^{new} < sup\ J$ и $sup\ J \in FP$, то суммарное количество FP-чисел в хороших интервалах уменьшилось при обработке.

Пусть вершина t_i определена. Тогда если интервал Y не содержит нуля, то уменьшение количеств FP-чисел среди хороших интервалов очевидно. Если $0 \in Y$ и $0 \notin Y^{'}$, то утверждение леммы также очевидно.

Пусть $0 \in Y^{'}$. Поскольку $inf\ x > inf\ X$, то интервал $X \cap [(inf\ X_2)^{-}, (sup\ X_2)^{+}]$ не является хорошим, а поскольку $X \cap [(inf\ X_1)^{-}, (sup\ X_1)^{+}] \subset X$, то утверждение леммы очевидно. Лемма доказана.

Для доказательства корректности алгоритма NC нам потребуется определить для каждой вершины-числа ее *характеристическую* функцию. Характеристические функции вершин-чисел — это функции вида $f : \overline{R} \rightarrow \overline{R}$, они будут определены ниже.

6.2. Вершины-функции

С каждой вершиной-функцией связана та же функция, что и в графе истории.

Каждая вершина-функция может быть *исполнена*. Пусть вершина-функция f имеет k входящих дуг. Исполнение вершины f — это следующее изменение связанных с ней значений или состояния “определена” или “не определена” вершины $t(f)$:

- если хотя бы одна из вершин $s_1(f), ..., s_k(f)$ не определена, то исполнение вершины f — это изменение состояния вершины $t(f)$ в значение “не определена”;
- пусть все вершины $s_1(f), ..., s_k(f)$ определены, и $F : \overline{R}^k \rightarrow \overline{R}$ — функция, связанная с вершиной f, $x_1, ..., x_k$ — точные значения, $X_1, ..., X_k$ — интервальные значения, а $X_1^{'}, ..., X_k^{'}$ — интервальные производные вершин $s_1(f), ..., s_k(f)$, то

  – если функция F определена, непрерывна и имеет частные производные первого порядка в многомерном интервале $X_1 \times ... \times X_k$, то исполнение вершины-функции f — это сохранение интервалов x_0, X_0 и $X_0^{'}$ таких, что
  $x_0 \supseteq F(x_1, ..., x_k)$
  $X_0 \supseteq F(X_1, ..., X_k)$

$X_0' \supseteq \sum_{i=1}^{k} F_i(X_1, ..., X_k)X_i'$
в качестве соответственно точного значения, интервального значения и интервальной производной вершины $t(f)$. При этом вершина $t(f)$ становится определенной. В этой записи F_i означает частную производную функции F по i-той переменной.

– в противном случае при исполнении вершины f вершина $t(f)$ также становится неопределенной.

6.3. Корректность графа-индикатора

Пусть I — стартовый интервал графа-индикатора IN.

Будем говорить, что граф IN корректен снизу, если после установки точного значения вершин-аргументов в $[inf\ I, inf\ I]$ и исполнения всех вершин-функций в порядке “старшие вперед” точные значения всех вершин-целей будут содержаться в $(0, \infty]$.

6.4. Предельное значение графа-индикатора

Пусть IN — граф-индикатор. *Стягиванием вниз* графа IN будем называть последовательное выполнение итераций, состоящих из следующих действий:

1) пусть $t_1, ..., t_n$ — все вершины-цели графа IN. Строим множество $S = \cup S_i$ — объединение наборов поисковых интервалов;

2) вычисляем значение $a = min\{inf\ X | X \in S\}$;

3) вычисляем значение $b = min\{sup\ X | X \in S\ \&\ a = inf\ X\}$;

4) строим список вершин целей $T = \{t_i | [a, b] \in S_i\}$;

5) строим список вершин функций $L = \{f_1, ..., f_s\}$, из которых достижима хотя бы одна вершина $t \in T$;

6) устанавливаем точное значение вершин-аргументов в $[(\frac{a+b}{2})^+, (\frac{a+b}{2})^+]$, значение интервала — в $[a, b]$, а интервальной производной — в $[1, 1]$;

7) исполняем все вершины из списка L в порядке "старшие вперед";

8) обрабатываем все вершины цели из списка T.

Стягивание вниз прекращается в случае, если или значения a и b, полученные на некоторой итерации, будут равны значениям, полученным на предыдущей итерации, или a станет больше b.

Нижним предельным значением графа в обоих случаях будем называть число a, полученное на шаге 2 последней итерации. Верхнее предельное значение определяется аналогично.

Теорема 2. Стягивание вниз графа-индикатора заканчивается за конечное количество итераций.

Доказательство. Предположим, что процесс стягивания продолжается бесконечно. Обозначим S^n — объединение наборов поисковых интервалов перед итерацией с номером n. Обозначим
$a^n = min\{inf\ X | X \in S^n\}$,
$b^n = min\{sup\ X | X \in S^n\ \&\ a^n = inf\ X\}$.

По лемме 7 для всех $i \geq 0$ верно: $a^i \leq a^{i+1}$. Поскольку все a^i — FP-числа, а множество FP — конечно, то $\exists m\ |\forall k\ k \geq m \Rightarrow a^k = a^m$. Обозначим Σ^n количество FP-чисел перед итерацией с номером n во всех поисковых интервалах I таких, что $inf\ I = a^m$. При этом, если какой-то интервал содержится в нескольких наборах поисковых интервалов, то количество его FP-чисел посчитано столько раз, в скольких интервалах он содержится.

По условию бесконечности процесса стягивания при всех k, больших m, верно: $b^k \neq b^{k+1}$. Поскольку все обработки вершин-целей по определению стягивания — правильные снизу, то по лемме 8, $\Sigma^k > \Sigma^{k+1}$ при всех k, больших m.

Следовательно, существует n такое, что $\Sigma^n = 0$. Тогда $a^n > b^n$, и приходим к противоречию.

6.5. Теорема о предельном значении графа-индикатора

Определим для каждой вершины-числа c графа-индикатора ее индикаторную функцию. Индикаторной функцией вершин-аргументов будет функция $f(x) = x$. Если c — вершина-константа, которая не является аргументом, a — значение, связанное с вершиной c, то индикаторной функцией вершины c будет функция $f(x) = a$. Если c — вершина-результат, вершина g такова, что $t(g) = c$, k — число входящих дуг вершины g, функции $f_1(x), ..., f_k(x)$ связаны с вершинами $s_1(g), ..., s_k(g)$, а функция $G(x_1, ..., x_k)$ связана с вершиной g, то индикаторной функцией вершины c будет функция $f(x) = G(f_1(x), ..., f_k(x))$. Индикаторные функции вершин-целей будем называть *целевыми* функциями.

Теорема 3. Пусть I — стартовый интервал графа индикатора IN, a — его нижнее предельное значение. Тогда если $a > inf\ I$, то все целевые функции графа IN определены, непрерывны и не имеют корней на интервале $[inf\ I, a)$.

Доказательство. Пусть f_i – индикаторная функция вершины-цели t_i.

Пусть $b \in [inf\ I, a)$. Пусть при некоторой обработке вершины t_i объединение ее поисковых интервалов перестало содержать точку b. Пусть x — точное значение, X — интервальное значение вершин-аргументов при этой обработке. Поскольку при обработке неопределенной вершины объединение ее поисковых интервалов не меняется, то вершина t_i определена. Следовательно, поскольку $b \in X$, то f_i непрерывна и имеет производную в точке b.

Пусть f_i имеет корень в точке $b \in [inf\ I, a)$. Пусть y — точное значение, Y — интервальное значение и $Y^{'}$ — интервальная производная вершины t_i во время упомянутой выше обработки вершины t_i.

Поскольку $b \in X$ — корень f_i, то по определению индикаторной функции и исполнения вершины-фунции интервал Y содержит ноль.

Поскольку b — корень f_i, то по теореме Лагранжа о среднем значении [1] существует точка c, которая лежит между точками b и $inf\ x$ такая, что $f_i^{'}(c)((inf\ x) - b) = f_i(inf\ x) - f_i(b)$. Но тогда $b \in x - y/Y^{'}$, следовательно, в результате обработки точка b остается в объединении поисковых интервалов вершины t_i. Приходим к противоречию, теорема доказана.

Следствие. Пусть I — стартовый интервал графа индикатора, a — его нижнее предельное значение. Тогда, если $a > inf\ I$ и граф корректен снизу, то все целевые функции графа индикатора положительны на интервале $[inf\ I, a)$.

7. АЛГОРИТМ NC

Алгоритм NC состоит из следующих этапов:

1) выбор главной границы;

2) создание графа задачи и графа истории;

3) пошаговое исполнение алгоритма M2B с сохранением информации в графе истории;

4) построение и стягивание графа-индикатора несовместности, уточнение главной границы.

Далее подробно описаны вышеперечисленные этапы.

7.1. Этап 1. Выбор главной границы

Пусть дана задача $M = (X, D, C)$. Выберем некоторую переменную $x_i \in X$ такую, что $inf\ D_i < sup\ D_i$, назовем ее *главной*. Выберем произвольную границу главной переменной $x \in \{inf\ D_i, sup\ D_i\}$ и назовем ее *главной границей.* Алгоритм NC будет сформулирован и доказан для случая, когда в качестве главной границы на этом этапе выбирается нижняя граница. Построим задачу $M^{'} = (X, D_1 \times ... \times D_{i-1} \times \{x\} \times D_{i+1} \times ... \times D_n, C)$.

На этом первый этап заканчивается.

7.2. Этап 2. Создание графа задачи и заготовки для графа истории

По задаче $M^{'}$ строим граф задачи G так, как это описывалось в разделе 2. Вершину-переменную графа G, соответствующую главной переменной задачи $M^{'}$, будем называть *главной вершиной-переменной.*

Далее на базе графа G строим граф истории H. После чего активизируем все вершины-отношения графа G, в которые идут дуги из главной вершины-переменной, и переходим к следующему этапу.

7.3. Этап 3. Пошаговое исполнение алгоритма M2B с сохранением информации в графе истории

Этап 3 состоит в последовательном исполнении итераций, одна итерация — это последовательное выполнение следующих действий:

1) выбор в графе G некоторой активной вершины отношения r. Вершину r будем называть *главной* вершиной итерации;

2) исполнение отношения r с сохранением следа в графе истории;

3) в случае, если значение вершины v изменилось, активизация всех вершин $r^{'}$ таких, что граф G содержит дугу $(v, r^{'})$;

4) деактивизация вершины r.

В случае, если в результате граф G станет пустым, переходим к следующему этапу. В случае, если граф G, оставаясь непустым, станет неактивным, работа алгоритма NC завершается неудачно. В этом случае мы можем вернуться к первому этапу, выбрать другую границу в качестве главной и запустить алгоритм сначала.

7.4. Этап 4. Построение графа-индикатора несовместности

Итак, на третьем этапе мы получили пустой граф задачи G и граф истории H, в котором сохранена информация о всех исполнениях ограничений в графе G. Пусть x_i — главная переменная задачи M, D_i — область значений переменной x в задаче M. На основании графа истории H и интервала D_i строим граф-индикатор несовместности IN.

Если граф не корректен снизу, применяем алгоритм M2B к задаче $(X, D_1 \times ... \times [inf\ D_i, inf\ D_i + \delta] \times ... \times D_n, C)$, где δ — некоторое маленькое число. В случае, если задача окажется несовместной, сужаем значение переменной x_i и повторяем алгоритм NC с той же главной границей. В противном случае алгоритм завершает свою работу неудачно.

Если граф корректен снизу, ищем его нижнее предельное значение. Пусть a — такое значение. Тогда, если $a = \infty$, задача M несовместна. В противном случае, если $D^{''} = D_1 \times ... \times D_{i-1} \times [a, sup\ D_i] \times D_{i+1} ... \times D_n$, то задача $(X, D^{''}, C)$ эквивалентна задаче M. Результатом работы алгоритма NC будет уточнение области возможных значений главной переменной: $D_i \to [a, sup\ D_i]$.

7.5. Теорема о корректности алгоритма NC

Теорема 4. Пусть дана задача $M = (X, D, C)$, где:
$X = \{x_1, ..., x_n\}$ — множество переменных;
$D = D_1 \times ... \times D_n$ — множество их возможных значений;
C — множество отношений.
Предположим, что к задаче M был применен алгоритм NC. Пусть нижняя граница переменной x_1 — главная граница NC алгоритма. Пусть G — граф задачи, H — граф истории, IN — граф-индикатор, которые были построены NC алгоритмом.

Предположим, алгоритм NC на последнем этапе получил значение a в качестве нижнего предельного значения графа-индикатора. Тогда, если $a = \infty$, задача M несовместна. В противном случае, если $D^{''} = [a, sup\ D_1] \times D_2 \times ... \times D_n$, то задача $(X, D^{''}, C)$ эквивалентна задаче M.

Доказательство. Обозначим $D_1^{'} = [inf\ D_1, a) \cap D_1$. Предположим, что $b = (b_1, ..., b_n) \in D_1^{'} \times D_2 ... \times D_n$ — решение задачи M.

Пусть множество T содержит все н-вершины и в-вершины графа IN.

Разобьем множество T на группы: если вершины f и g — ровесники, то они входят в одну группу, иначе — в разные. Поскольку на каждой итерации алгоритма NC может быть создано не более одной н-вершины и не более одной в-вершины, то каждая группа содержит либо одну, либо две вершины.

Пусть m — количество получившихся групп. Пронумеруем группы числами $1, ..., m$ так, чтобы группы, в которые входят более старшие вершины, имели меньший порядковый номер. Если j — номер некоторой группы, то номер итерации алгоритма NC, на которой были созданы вершины из этой группы, будем обозначать m_j. Пусть также $m_0 = 0$.

Пусть j — номер некоторой группы, вершина r графа задачи — главная вершина итерации m_j, вершина $t(r)$ соответствует переменной x_l задачи M. Тогда будем говорить, что j-тая группа *соответствует* переменной x_l.

Определим последовательность n-мерных замкнутых интервалов $I^0, ..., I^m$, границы которых принадлежат множеству $\overline{R}$.

Пусть $I^0 = [b_1, b_1] \times D_2 \times ... \times D_n$. Если $1 \leq j \leq m$ и j-тая группа множества T соответствует переменной x_k, $1 \leq k \leq n$, то, если она содержит некоторую н-вершину f, обозначим индикаторную функцию вершины $t(f)$ как F, а если она содержит некоторую в-вершину g, обозначим индикаторную функцию вершины $t(g)$ как G. Тогда:

$$I_s^j = \begin{cases} I_s^{j-1}, & \text{если } s \neq k; \\ [F(b_1), G(b_1)], & \text{если } s = k \text{ и } j\text{-тая группа содержит} \\ & \text{н- и в-вершины;} \\ [F(b_1), sup\ I_s^{j-1}], & \text{если } s = k \text{ и } j\text{-тая группа содержит только} \\ & \text{н-вершину;} \\ [inf\ I_s^{j-1}, G(b_1)], & \text{если } s = k \text{ и } j\text{-тая группа содержит только} \\ & \text{в-вершину.} \end{cases}$$

Лемма 9. Пусть вершина-переменная v_k соответствует некоторой переменной x_k, $1 \leq k \leq n$. Пусть граф-индикатор содержит вершину c, при этом либо $c = l_s(v_k)$, либо $c = h_s(v_k)$, где $1 \leq s \leq m_m$. Обозначим $j = max\{q|s \geq m_q\}$. Пусть F — индикаторная функция вершины c. Тогда, если $c = l_s(v_k)$, то $inf\ I_k^j = F(b_1)$, а если $c = h_s(v_k)$, то $sup\ I_k^j = F(b_1)$.

Доказательство. Докажем лемму для случая $c = l_s(v_k)$. Пусть c — вершина-граница. Тогда $time(c) = 0$. По определению заготовки графа истории значение вершины c равно $sup\ D_k$. Если c — вершина-аргумент, тогда $k = i$ и $F(b_1) = b_1 = inf\ I_1^0 = inf\ I_k^0$. В противном случае, $D_k = I_k^0$ и $F(b_1) = inf\ D_k$. Следовательно, также верно равенство: $F(b_1) = inf\ I_k^0$. Осталось показать, что $sup\ I_k^j = inf\ I_k^0$.

Пусть $infI_k^j \neq infI_k^0$, тогда существует как минимум одна группа множества T с номером, не большим j, которая соответствует переменной x_k и содержит некоторую н-вершину. Тогда по лемме 6 $time(l_s(v_k)) \geq m_j$, противоречие. Таким образом, если c — вершина-граница, то $F(b_1) = infI_k^j$.

Если c — вершина-результат, тогда в графе-индикаторе существует н-вершина g такая, что: $t(g) = c$. Пусть с g связана функция $\widehat{G}$. Пусть $time(c) = m_{j_1}$, поскольку $c = l_s(v_k)$, то $m_{j_1} \le s$. По построению последовательности $inf\ I_k^{j_1} = F(b_1)$. Осталось показать, что $inf\ I_k^j = inf\ I_k^{j_1}$.

Пусть $inf\ I_k^j \neq inf\ I_k^{j_1}$, тогда существует как минимум одна группа множества T с номером из диапазона $[j_1 + 1, j]$, которая соответствует переменной x_k и содержит некоторую н-вершину. Тогда по лемме 6 $time(l_s(v_k)) \ge m_{j_1+1}$, получаем противоречие. Таким образом, если c — вершина-результат, то $F(b_1) = inf\ I_k^j$.

Лемма доказана.

Продолжим доказательство основной теоремы. Предположим, что для некоторого j, где $1 \le j \le m$, верно: $b \in I^{j-1}$. Докажем, что $b \in I^j$.

Предположим, что j-тая группа содержит единственную вершину — н-вершину f. Пусть r — вершина-отношение, исполнение которой на итерации m_j характеризует вершина f. Пусть она соответствует отношению $c_l(x_{l_1}, ..., x_{l_u})$.

Пусть набор вершин $f_1, ..., f_t$ — набор условий корректности вершины f. Тогда граф истории по построению содержит дуги $(t(f_k), t(f))$, $1 \le k \le t$. Следовательно, вершины $f_1, ..., f_t$ содержатся в графе IN.

Пусть $\widehat{F}, \widehat{F}_1, ..., \widehat{F}_t$ — функции, связанные с вершинами $f, f_1, ..., f_t$, а $F, F_1, ..., F_{t_1}$ — усложнения соответственно функций $\widehat{F}, \widehat{F}_1, ..., \widehat{F}_t$. Пусть $(p_1, ..., p_s)$ — вектор параметров функции F.

Обозначим $y = (inf\ I_{l_1}^{j-1}, sup\ I_{l_1}^{j-1}, ..., inf\ I_{l_u}^{j-1}, sup\ I_{l_u}^{j-1}, p_1, ..., p_s)$. Пусть G — индикаторная функция вершины $t(f)$. Покажем, что $F(y) = G(b_1)$.

Пусть $e_1, ..., e_w$, $1 \le e_1 < ... < e_w \le 2u + s$ — набор индексов такой, что $F(y_1, ..., y_{2u+s}) = \widehat{F}(y_{e_1}, ..., y_{e_w})$. Обозначим $\widehat{y} = (y_{e_1}, ..., y_{e_w})$.

Пусть вершина $c_k = s_k(f)$, $1 \le k \le w$, заметим, что она входит в граф-индикатор. По построению графа истории c_k это:

– вершина-параметр со связанным значением $p_{e_k - 2u}$, если $e_k > 2u$;

– $h_{m_j - 1}(v_{\frac{e_k+1}{2}})$, если $e_k \le 2u$ и e_k — четное;

– $l_{m_j - 1}(v_{\frac{e_k}{2}})$, если $e_k \le 2u$ и e_k — нечетное.

Пусть G_k — индикаторная функция вершины c_k. Вычислим значение $G_k(b_1)$.

Пусть $e_k > 2u$, тогда c_k — вершина-параметр, ее индикаторная функция: $G_k(x) = p_{e_k - 2u}$. Следовательно, $G_k(b_1) = \widehat{y}_k$.

Пусть $e_k \le 2u$ и число e_k — четное. Тогда $c_k = h_{m_j - 1}(v_{\frac{e_k+1}{2}})$. Обозначим $s = max\{q|m_j - 1 \geq m_q\}$, тогда по лемме 9 $sup\ I_k^s = G_k(b_1)$. Поскольку $max\{q|m_j - 1 \geq m_q\} = m_{j-1}$, то $sup\ I_k^s = sup\ I_k^{j-1} = \widehat{y}_k$.

Аналогично, если $e_k \leq 2u$ и число e_k — нечетное, то $G_k(b_1) = \widehat{y}_k$.

Таким образом, для всех k из диапазона $1 \leq k \leq w$ верно $G_k(b_1) = \widehat{y}_k$. По определению индикаторной функции получаем $G(b_1) = \widehat{F}(\widehat{y})$. Следовательно, $G(b_1) = F(y)$. Аналогично для вершин $f_1, ..., f_t$ получаем, что, если G_k — индикаторная функция вершины $t(f_k)$, то $G_k(b_1) = F_k(y)$. Поскольку вершины $f_1, ..., f_t$ являются целями и $b_1 \in [inf\ D_1, a)$, то по следствию из теоремы 3 верно: $G_k(b_1) > 0$, при $1 \leq k \leq t$.

Пусть переменная x_k связана с j-той группой множества T, тогда по определению последовательности $I^0, ..., I^m$ $inf\ I_k^j = G(b_1)$. По определению набора условий корректности $inf\ I_k^j \leq inf\ P_k(c_l \cap I_{l_1}^{j-1} \times ... \times I_{l_u}^{j-1})$. Поскольку b — решение задачи M, то $(b_{l_1}, ..., b_{l_u}) \in c_l$, следовательно, $inf\ I_k^j \leq b_k$. По определению последовательности $I^0, ..., I^m$ $sup\ I_k^j = sup\ I_k^{j-1}$ и $I_s^j = I_s^{j-1}$ для всех $s \neq k$. Следовательно, поскольку $b \in I^{j-1}$, то $b \in I^j$.

Если j-тая группа содержит только в-вершину или н- и в- вершины, то аналогично доказывается, что $b \in I^j$. Таким образом, доказано, что, если $b \in I^{j-1}$, то $b \in I^j$. Поскольку $b \in I^0$, то по индукции получаем: $b \in I^m$.

Пусть v_k — пустая вершина графа-индикатора. Пусть она соответствует переменной x_k задачи M. Обозначим F_1 — индикаторную функцию вершины $l_{m_m}(v_k)$, а F_2 — индикаторную функцию вершины $h_{m_m}(v_k)$. По лемме 9 $inf\ I_k^m = F_1(b_1)$, а $sup\ I_k^m = F_2(b_1)$.

Пусть c_0 — главная вершина-цель графа IN. Пусть F_0 — индикаторная функция вершины c_0, тогда по построению графа-индикатора $F_0(x) = F_1(x) - F_2(x)$. Поскольку c_0 — вершина-цель, граф IN корректен снизу, $b_1 \in [inf\ D_1, a)$, то по следствию из теоремы 3 $F_0(b_1) > 0$, следовательно, $inf\ I_k^m > sup\ I_k^m$, отсюда $I^m = \emptyset$. Это противоречит тому, что $b \in I^m$, значит теорема доказана.

8. СРАВНЕНИЕ СЛОЖНОСТИ АЛГОРИТМОВ NC И M2B НА ПРИМЕРЕ КОНКРЕТНОЙ ЗАДАЧИ

Пусть дана следующая система уравнений:
$x_1^2 = x_2$,
$x_2 + 1 = x_3$,
$x_3 = 2x_1$.

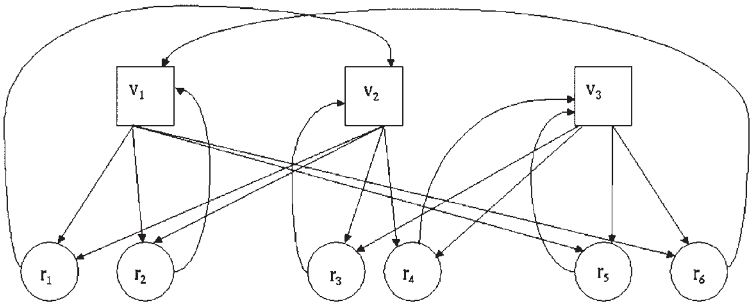

Рис. 1. Граф задачи

Построим по ней численную задачу удовлетворения ограничений $M = (X, D, C)$, где:

$X = \{x_1, x_2, x_3\}$,
$D = D_1 \times D_2 \times D_3$,
$D_1 = [0.5, 1]$,
$D_2 = [0, \infty]$,
$D_3 = [0, \infty]$,
$C = \{c_1(x_2, x_1), c_2(x_2, x_3), c_3(x_3, x_1)\}$,
$c_1 = \{(x, y) | x = y^2\}$,
$c_2 = \{(x, y) | x = y + (-1)\}$,
$c_3 = \{(x, y) | x = 2y\}$.

Теперь опишем решение полученной задачи алгоритмами M2B и NC.

8.1. Решение задачи M алгоритмом M2B

Граф задачи G, построенный по M, будет содержать три вершины-переменные v_1, v_2 и v_3; шесть вершин-отношений: r_1, r_2 будут соответствовать c_1; r_3, r_4 будут соответствовать c_2; r_5, r_6 будут соответствовать c_3. Входящие дуги каждой вершины-отношения однозначно определены переменными, которые связывает соответствующее отношение задачи M. Выходящие дуги будут следующие: (r_1, v_2), (r_2, v_1), (r_3, v_2), (r_4, v_3), (r_5, v_3), (r_6, v_1).

Рассмотрим работу алгоритма M2B на графе G. От порядка выбора вершин для исполнения зависит время работы алгоритма M2B, поэтому в различных реализациях алгоритма M2B применяются различные эвристики для выбора активной вершины. Сначала для простоты рассмотрим работу алгоритма при условии, что для исполнения выбирается активная вершина-отношение с минимальным индексом, а затем оценим, каким образом можно оптимизировать его работу.

Предположим, что на некоторой итерации алгоритма M2B исполнилась вершина r_1. Опустим подробности, скажем лишь, что перед тем, как вершина r_1 будет исполнена следующий раз, будут последовательно исполнены вершины

$$r_2,\ \mathbf{r_3},\ r_4,\ r_5,\ \mathbf{r_6}.$$

Выделенным шрифтом отмечены те вершины, исполнение которых влечет изменение значения некоторой вершины-переменной.

Итак, если для исполнения выбирается активная вершина с минимальным индексом, то итерации алгоритма M2B можно разбить на группы, каждая из которых содержит итерации, которые последовательно исполняют вершины

$$r_1,\ r_2,\ r_3,\ r_4,\ r_5,\ r_6.$$

При этом исполнение вершин r_2, r_4 и r_5 не будет приводить к изменениям значений вершин-переменных.

Предположим, что некоторый эвристический алгоритм выбирает последовательно вершины r_1, r_3, r_6, и только, если все они станут неактивными, выбирает одну из вершин r_2, r_4, r_5. В этом случае работа алгоритма M2B сведется к последовательному исполнению вершин

$$r_1,\ r_3,\ r_6,\ r_1,\ \ldots\ .$$

Несложно доказать, что при таком выборе вершины для исполнения время работы M2B будет минимальным.

Предположим, что после некоторого исполнения вершины r_1 значение, связанное с вершиной v_1, стало $[1-\delta, 1]$, где $\delta < 1$. Тогда после следующего исполнения вершины r_1 значение, связанное с вершиной v_1, станет $[1-\delta+\frac{\delta^2}{2}, 1]$.

В случае, если множество FP -это множество 64-разрядных чисел с мантиссой 52 двоичных знака, то через 158915040 итераций значение вершины v_1 станет равным $[a, 1]$, где $a \in FP$, таково, что $(\frac{((a^2)^- + 1)^-}{2})^- = a$, a приблизительно равно $1 - 1.49 \cdot 10^{-8}$.

8.2. Решение задачи M алгоритмом NC

8.2.1. Этап 1. Выбор главной границы

Итак, пусть нижняя граница переменной x_1 будет главной. Построим задачу $M' = (X, [0.5, 0.5] \times D_2 \times D_3, C)$ и переходим ко второму этапу.

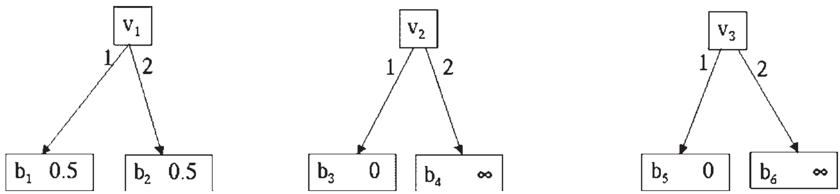

Рис. 2. Заготовка графа истории

8.2.2. Этап 2. Создание графа задачи и заготовки для графа истории

По задаче $M^{'}$ строим граф $G^{'}$. Поскольку главной является нижняя граница переменной x_1, граф $G^{'}$ будет отличаться от графа G, построенного в предыдущем разделе, только значением, связанным с вершиной v_1. Значение вершины v_1 в графе $G^{'}$ — интервал $[0.5, 0.5]$.

На базе графа $G^{'}$ строим заготовку графа истории H. Она будет содержать 3 вершины-переменные v_1, v_2 и v_3, с которыми соответственно связаны значения $[0.5, 0.5]$, $[0, \infty]$ и $[0, \infty]$; шесть вершин-границ b_1, ..., b_6, с которыми соответственно связаны значения 0.5, 0.5, 0, ∞, 0 и ∞; и дуги (v_i, b_{2i-1}), (v_i, b_{2i}), $i = 1, ..., 3$. Причем дуги (v_i, b_{2i-1}) будут первыми выходящими дугами вершин v_i, $i = 1, ..., 3$. Вершины b_1 и b_2 будут аргументами.

Далее активизируем все вершины-отношения графа $G^{'}$ и переходим к третьему этапу.

8.2.3. Этап 3. Пошаговое исполнение алгоритма M2B с сохранением информации в графе истории

Здесь мы также будем выбирать для исполнения активную вершину-отношение с минимальным индексом.

Итерация 1. r_1 — главная вершина итерации. Исполняем r_1: вычисляем значение $A = P_1([0.5, 0.5] \times [0, \infty] \cap \{(x, y) | x = y^2)$. Получаем $A = [0.5, 0.5]$, значение вершины v_1 не меняется. Делаем не активной вершину r_1.

Итерация 2. r_2 — главная вершина итерации. Исполняем r_2: вычисляем значение $A = P_2([0.5, 0.5] \times [0, \infty] \cap \{(x, y) | x = y^2)$. Получаем $A = [0.25, 0.25]$, поскольку $0.25 \in FP$, то новое значение вершины v_2 — интервал $[0.25, 0.25]$.

В качестве нижней характеристики исполнения r_1 возьмем функцию

$F_1(y_1, y_2, y_3, y_4) = y_1^2$. Ее упрощение — это функция $\widehat{F}_1(y) = y^2$. Сохраняем характеристику исполнения r_1 для нижней границы значения v_2:

1) создаем вершину-результат c_1;
2) создаем н-вершину f_1 со связанной функцией: $\widehat{F}_1$;
3) создаем дуги (b_1, f_1) и (f_1, c_1).

В качестве верхней характеристики исполнения r_1 возьмем функцию $F_2(y_1, y_2, y_3, y_4) = y_2^2$. Ее упрощение — это функция $\widehat{F}_2(y) = y^2$. Сохраняем характеристику исполнения r_1 для верхней границы значения v_2:

1) создаем вершину-результат c_2;
2) создаем в-вершину f_2 со связанной функцией: $\widehat{F}_2$;
3) создаем дуги (b_2, f_2) и (f_2, c_2).

Набор условий корректности для функции F_1 будет состоять из единственной функции $F_3(y_1, y_2, y_3, y_4) = y_1$, Действительно, если $y \in [y_1, y_2]$ и $y_1 > 0$, то $y^2 \geq y_1^2$. Упрощение функции F_3 — функция $\widehat{F}_3(y) = y$. Сохраняем условие корректности для f_1:

1) создаем вершину цель t_1;
2) создаем вершину-условие f_3 со связанной функцией исполнения $\widehat{F}_3$;
3) создаем дуги (b_1, f_3), (f_3, t_1) и (t_1, c_1).

Набор условий корректности для функции F_2 будет состоять из функций $F_4(y_1, y_2, y_3, y_4) = y_2$ и $F_5(y_1, y_2, y_3, y_4) = y_1 + y_2$. Действительно, если $y \in [y_1, y_2]$, $y_2 > 0$ и $y_1 + y_2 > 0$, то $y^2 \leq y_2^2$. Упрощение F_4 — функция $\widehat{F}_4(y) = y$, упрощение F_5 — функция $\widehat{F}_5(y_1, y_2) = y_1 + y_2$. Сохраняем условия корректности для f_2:

1) создаем вершину-цель t_2;
2) создаем вершину-условие f_4 со связанной функцией исполнения $\widehat{F}_4$;
3) создаем дуги (b_2, f_4), (f_4, t_2) и (t_2, c_2).

А также:

1) создаем вершину-цель t_3;
2) создаем вершину-условие f_5 со связанной функцией исполнения $\widehat{F}_5$;
3) создаем дуги (b_1, f_5), (b_2, f_5), (f_5, t_3) и (t_3, c_2).

Выполняем преобразования $\underline{trace}(H, v_2, c_1)$ и $\overline{trace}(H, v_2, c_2)$:

1) заменяем дугу (v_2, b_3) дугой (v_2, c_1);
2) заменяем дугу (v_2, b_4) дугой (v_2, c_2).

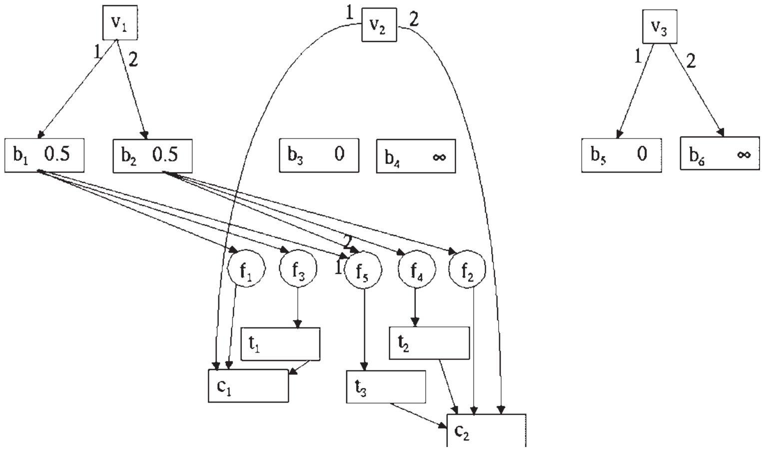

Рис. 3. Граф истории после второй итерации

Делаем активными вершины r_1, r_2, r_6, r_7 и r_8. Делаем неактивной вершину r_2.

Итерация 3. r_1 — главная вершина итерации. Исполняем r_1: вычисляем значение $A = P_1([0.5, 0.5] \times [0.25, 0.25] \cap \{(x, y)|x = y^2)$. Получаем $A = [0.5, 0.5]$, значение вершины v_1 не меняется. Делаем не активной вершину r_1.

Итерация 4. r_3 — главная вершина итерации. Исполняем r_3: вычисляем значение $A = P_2([0.25, 0.25] \times [0, \infty] \cap \{(x, y)|x = y - 1)$. Получаем $A = [1.25, 1.25]$, поскольку $1.25 \in FP$, то новое значение вершины v_3 — интервал $[1.25, 1.25]$.

В качестве нижней характеристики исполнения r_3 возьмем функцию $F_6(y_1, y_2, y_3, y_4, y_5) = y_1 + y_5$ с вектором параметров, состоящим из единственного элемента — числа 1. Ее упрощение — это функция $\widehat{F}_6(y_1, y_2) = y_1 + y_2$. Сохраняем характеристику исполнения r_3 для нижней границы значения v_3:

1) создаем вершину-параметр p_1 со значением 1;
2) создаем вершину-результат c_3;
3) создаем н-вершину f_6 со связанной функцией $\widehat{F}_6$;
4) создаем дуги (c_1, f_6), (p_1, f_6) и (f_6, c_3).

В качестве верхней характеристики исполнения r_3 возьмем функцию $F_7(y_1, y_2, y_3, y_4, y_5) = y_2 + y_5$ с вектором параметров, состоящим из единственного элемента — числа 1. Ее упрощение — это функция

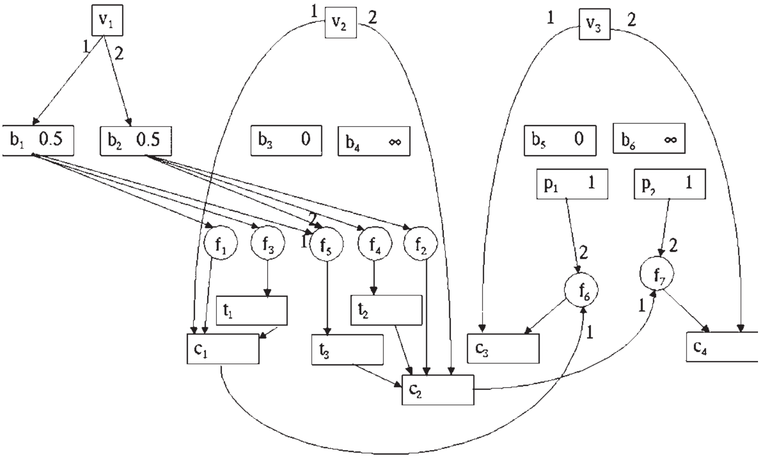

Рис. 4. Граф истории после четвертой итерации

$\widehat{F}_7(y_1, y_2) = y_1 + y_2$. Сохраняем характеристику исполнения r_3 для верхней границы значения v_3:

1) создаем вершину-параметр p_2 со значением 1;
2) создаем вершину-результат c_4;
3) создаем в-вершину f_7 со связанной функцией $\widehat{F}_7$;
4) создаем дуги (c_2, f_7), (p_2, f_7) и (f_7, c_4).

Наборы условий корректности для функций F_6 и F_7 будут пустыми.

Выполняем преобразования $\underline{trace}(H, v_3, c_3)$ и $\overline{trace}(H, v_3, c_4)$:

1) заменяем дугу (v_3, b_5) дугой (v_3, c_3);
2) заменяем дугу (v_3, b_6) дугой (v_3, c_4).

Делаем активными вершины r_3, ..., r_8. Делаем неактивной вершину r_3.

Итерация 5. r_4 — главная вершина итерации. Исполняем r_4: вычисляем значение $A = P_1([0.25, 0.25] \times [1.25, 1.25] \cap \{(x, y)|x = y - 1)$. Получаем $A = [0.25, 0.25]$, значение вершины v_3 не меняется. Делаем не активной вершину r_4.

Итерация 6. r_5 — главная вершина итерации. Исполняем r_5: вычисляем значение $A = P_1([1.25, 1.25] \times [0.5, 0.5] \cap \{(x, y)|x = 2y)$. Получаем $A = \emptyset$, новое значение вершины v_3 — интервал $[\infty, -\infty]$.

В качестве нижней характеристики исполнения r_5 возьмем функцию

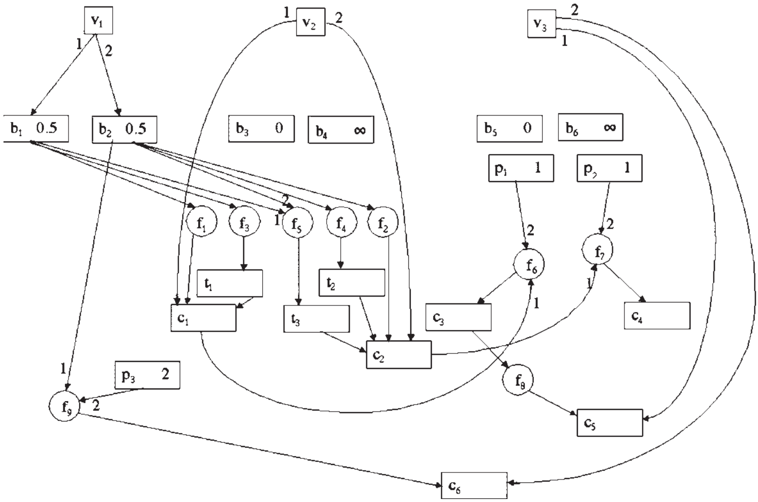

Рис. 5. Граф истории после шестой итерации

$F_8(y_1, y_2, y_3, y_4) = y_1$. Ее упрощение — это функция $\widehat{F}_8(y) = y$. Сохраняем характеристику исполнения r_5 для нижней границы значения v_3:

1) создаем вершину-результат c_5;
2) создаем н-вершину f_8 со связанной функцией $\widehat{F}_8$;
3) создаем дуги (c_3, f_8) и (f_8, c_5).

В качестве нижней характеристики исполнения r_5 возьмем функцию $F_9(y_1, y_2, y_3, y_4, y_5) = y_4 y_5$ с вектором параметров, состоящим из единственного элемента — числа 2. Ее упрощение — это функция $\widehat{F}_9(y_1, y_2) = y_1 y_2$. Сохраняем характеристику исполнения r_5 для верхней границы значения v_3:

1) создаем вершину-параметр p_3 со связанным с ней значением 2;
2) создаем вершину-результат c_6;
3) создаем в-вершину f_9 со связанной функцией: $\widehat{F}_9$;
4) создаем дуги (b_2, f_9), (p_3, f_9) и (f_9, c_6).

Наборы условий корректности для функций F_8 и F_9 будут пустыми.

Выполняем преобразования $\underline{trace}(H, v_3, c_5)$ и $\overline{trace}(H, v_3, c_6)$:

1) заменяем дугу (v_3, c_3) дугой (v_3, c_5);
2) заменяем дугу (v_3, c_4) дугой (v_3, c_6).

Переходим к следующему этапу.

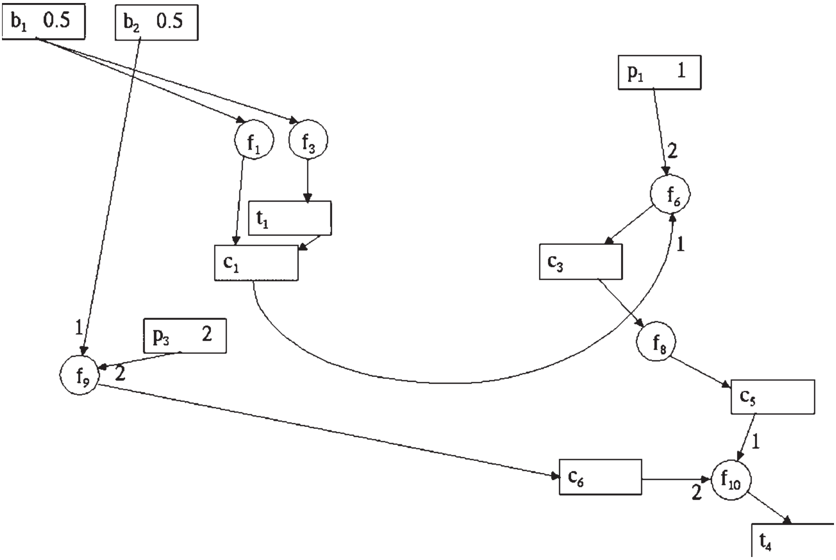

Рис. 6. Граф индикатор

8.2.4. Этап 4. Построение графа-индикатора несовместности

Итак, мы получили пустой граф истории. Значение вершины v_3 пусто. Добавим в граф истории вершину-цель t_4, вершину-условие f_{10} со связанной с ней функцией $\widehat{F}_{10}(y_1, y_2) = y_1 - y_2$ и дуги (c_5, f_{10}) и (c_6, f_{10}).

Построим граф индикатор, как описывалось выше. В него войдут:

1) вершины-цели t_1, t_4;
2) вершины-результаты (не цели) c_1, c_3, c_5, c_6;
3) вершины-аргументы b_1, b_2;
4) вершины-параметры p_1, p_3;
5) н-вершины f_1, f_6, f_8;
6) в-вершина f_9;
7) вершины-условия f_3, f_{10};
8) все дуги графа H, которые входят или выходят из упомянутых вершин-функций.

С вершинами t_1, t_4 связаны соответственно множества S_1, S_4. Изначально $S_1 = S_4 = \{[0.5, 1]\}$.

Проверяем корректность графа IN снизу. Устанавливаем значение вершин-аргументов в 0.5 и исполняем вершины f_1, f_3, f_6, f_8, f_9, f_{10}. В результате значение вершины t_1 будет $[0.25, 0.25]$, а вершины t_4 — $[0.25, 0.25]$. Граф корректен снизу.

Приступаем к стягиванию графа-индикатора.

Итерация 1.

1) $S = \cup S_i = \{[0.5, 1]\}$;
2) $a = 0.5$;
3) $b = 1$;
4) $T = \{t_1, t_4\}$;
5) $L = \{f_1, f_3, f_6, f_8, f_9, f_{10}\}$;
6) устанавливаем точное значение вершин b_1 и b_2 в $[0.75, 0.75]$, значение интервала в $[0.5, 1]$, интервальной производной в $[1, 1]$;
7) исполняем все вершины из списка L;
8) обрабатываем вершины t_1, t_4; получаем $S_1 = \emptyset$, $S_4 = \{[0.8125, 1]\}$.

Итерация n. $2 \leq n \leq 18$.

1) $S = \cup S_i = \{[a_n, 1]\}$;
2) $a = a_n$;
3) $b = 1$;
4) $T = \{t_4\}$;
5) $L = \{f_1, f_6, f_8, f_9, f_{10}\}$;
6) устанавливаем точное значение вершин b_1 и b_2 в $[(\frac{a_n+1}{2})^+, (\frac{a_n+1}{2})^+]$, значение интервала в $[a_n, 1]$, интервальной производной в $[1, 1]$;
7) исполняем все вершины из списка L;
8) обрабатываем вершину t_4; получаем: $S_4 = \{[a_{n+1}, 1]\}$.

При этом a_{n+1} вычисляется по формуле $a_{n+1} = (x - (\frac{((x^2+1)^- - (2x)^+)^-}{(2a-2)^-})^+)^-$, где $x = (\frac{a+b}{2})^+$, с выполнением корректно направленных округлений [10]. Если вычисления производились бы с бесконечной точностью, выполнялось бы $\delta_{n+1} = \frac{3\delta_n}{8}$, где $\delta_n = 1 - a_n$.

В реальности, если множество FP — это множество 64-разрядных вещественных чисел, то, начиная с 19-й итерации, точное значение вершины t_4 будет содержать ноль, ее обработка будет приводить к увеличению количества поисковых интервалов, и на 46-й итерации в качестве значения b мы получим то же значение, что и на 45-й итерации, при этом a и b будут соседними FP-числами. Алгоритм закончит свою работу со значением a, приблизительно равным $1 - 1.12181888 \cdot 10^{-8}$.

Таким образом, мы получили лучший результат за гораздо меньшее количество выполненных действий.

9. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

В этом разделе представлены экспериментальные результаты сравнения предлагаемого в статье алгоритма с другими существующими алгоритмами.

Алгоритм M2B был описан выше. Алгоритм Split описан в [7] и является комбинацией алгоритмов 2B и бисекции. Причем под Split-1 имеется ввиду поиск первого корня при помощи алгоритма Split, а под Split-all — поиск и объединение всех корней при помощи алгоритма Split. Алгоритм SC [2] является улучшением стандартного алгоритма достижения более сильной локальной совместности — так называемой *3B-совместности* [5].

В приведенной ниже таблице представлено время работы (в секундах) и результаты работы алгоритмов на примерах кубического уравнения:

$x^3 - 3x^2 + 3x - 1 = 0$ (первая строка таблицы)

и задачи Broyden Banded [9] для $n = 20$:

$x_i(2 + 5x_i^2) = -1 + \sum_{k \in I_i} x_k(1 + x_k)$; $i = 1, .., n$,

где $I_i = [max\{1, i - 5\}, min\{n, i + 1\}] \backslash \{i\}$ (вторая и третья строки таблицы).

Начальная область	M2B	Split-1	Split-all	SC	NC
$[-1 \cdot 10^8, 1 \cdot 10^8]$	< 1 с 5.713	96 с $1 \cdot 10^{-16}$	> 1 сут	696 с $3.98 \cdot 10^{-3}$	< 1 с $1.25 \cdot 10^{-4}$
$[-1 \cdot 10^8, 1 \cdot 10^8]^{20}$	< 1 с $2 \cdot 10^8$	69 с $1 \cdot 10^{-8}$	> 1 сут	10748 с $1 \cdot 10^{-8}$	73 с $1 \cdot 10^{-8}$
$[-1, 1]$	< 1 с 2	233 с $1 \cdot 10^{-8}$	> 1 сут	20 с $1 \cdot 10^{-8}$	63 с $1 \cdot 10^{-8}$

Приведенные результаты получены на Sun Ultra-60.

В связи с большим количеством накладных расходов алгоритма NC его эффективность заметна при использовании с большими начальными областями.

В настоящее время все большее распространение получает идея создания так называемых *кооперативных решателей* [11, 8], в которых на разных этапах решения задачи применяются различные алгоритмы. Применение представленного алгоритма NC в качестве одного из инструментов кооперативного решателя существенно ускоряет решение задачи в целом. Среди недостатков алгоритма стоит отметить большой расход памяти при решении задач с большим количеством переменных.

10. ЗАКЛЮЧЕНИЕ

В данной работе предложен разработанный автором алгоритм NC. Алгоритм предназначен для уточнения существующей внешней оценки решения. Алгоритм основан на построении и стягивании графа-индикатора несовместности некоторой подзадачи исходной задачи. Алгоритм может быть применен как для решения квадратных систем уравнений, так и для решения систем, количество переменных которых не совпадает с количеством уравнений. Результаты приведенных экспериментов показывают значительное преимущество предлагаемого алгоритма перед используемыми в настоящее время.

СПИСОК ЛИТЕРАТУРЫ

1. **Выгодский М. Я.** Справочник по высшей математике. — М: Наука, 1975. — С. 330–331.
2. **Лоенко М. Ю.** Решение систем нелинейных уравнений методами интервального распространения ограничений // Вычислительные технологии. — (В печати).
3. **Hansen E.** Global optimization using interval analysis. — N.-Y.: Marcel Dekker, 1992.
4. **Kashevarova T., Leshchenko A., Petunin D., Semenov A.** Combining various techniques with the algorithm of subdefinite calculations // Proc. of the 3rd Intern. Conf. of PACT'97. — London, England, (April, 1997). — P. 287–306.
5. **Lhomme O.** Consistency techniques for numeric CSP's // Proc. of the 13th IJCAI / Ed. by R. Bajcsy. — IEEE Computer Society Press, 1993. — P. 232–238.
6. **Lhomme O., Gotlieb A., Rueher M.** Dynamic optimization of Interval Narrowing Algorithms // J. of Logic Programming. — 1998. — Vol. 37, N 1-3. — P. 165—183
7. **Loyenko M.** Solving systems of nonlinear equations with methods using interval constraint propagation // Proc. of Intern. Sympos. on Scientific Computing, Computer Arithmetic and Validated Numerics SCAN-98 (extended abstracts), Budapest, September 22–25, 1998. — P. 98–99.
8. **Marti P., Rueher M.** A distributed cooperating constraints solving system // Intern. J. on Artificial Intelligence Tools. — 1995. — Vol. 4, N 1—2. — P. 93—113. (ps available from: http://wwwi3s.unice.fr/ rueher/)
9. **More J. J., Garbow B. S., Hillstrom K. E.** Testing Unconstrained Optimization Software //ACM Trnsactions on Mathematical Software. — 1981. — Vol. 7, N1. — P. 17–41.
10. **Numerical** Computation Guide. — Mountain View, USA, November, 1995.
11. **Rueher M.** An architecture for cooperating constraint solvers on reals //Constraint Programming: Basics and Trends. — Berlin a.o.: Springer Verlag, 1995. — P.231–250. — (Lect. Notes Comput. Sci.; Vol. 910).

М. Ю. Лоенко

УЛУЧШЕНИЕ ВНЕШНЕЙ ОЦЕНКИ МНОЖЕСТВА РЕШЕНИЙ ЗАДАЧ УДОВЛЕТВОРЕНИЯ ОГРАНИЧЕНИЙ

Препринт
79

Рукопись поступила в редакцию 12.10.2000
Рецензент В. А. Евстигнеев
Редактор З. В. Скок

Подписано в печать 20.12.2000
Формат бумаги 60×84 1/16 Объем 2,4 уч.-изд.л., 2,6 п.л.
Тираж 50 экз.

НФ ООО ИПО "Эмари" РИЦ, 630090, г. Новосибирск, пр. Акад. Лаврентьева, 6