

УДК 681.3

## Разработка алгоритма интеллектуальной поддержки улучшения промежуточных решений оптимизационных задач\*

Ю. О. Чернышёв, Н. Н. Венцов

(Донской государственный технический университет),

С. А. Мухтаров

(Филиал Военной академии связи (г. Краснодар))

*Рассмотрены проблемные области, требующие интеллектуальной поддержки процесса решения оптимизационных задач. Разработан адаптивный алгоритм поддержки решения оптимизационных задач на основе лингвистических команд. Командам могут сопоставляться как чёткие, так и расплывчатые характеристики параметров вычислительного процесса.*

**Ключевые слова:** адаптация, интеллектуальные системы, оптимизация.

**Введение.** Задачи оптимизации занимают важное место в современных информационных системах: поддержки принятия решений, автоматизации проектирования, моделирования бизнес-процессов. В ряде случаев наиболее эффективным подходом к решению оптимизационных задач является конкатенация вычислительной системы и лица, принимающего решения (ЛПР). Задачи, требующие подобного подхода, часто встречаются при реализации маршрутов проектирования СБИС. По причине ограниченности вычислительных ресурсов топология схемы не может быть спроектирована в целом, в связи с чем уже на начальных этапах проектирования прибегают к декомпозиции (разбиению на сегменты) проектируемой схемы. Проектирование каждого сегмента, как правило, связано с решением в том числе и ряда оптимизационных задач. В качестве примера рассмотрим задачи о вершинном покрытии и размещении.

Множество вершин  $X_v \subseteq X$  графа  $G = (X, U)$  называется *вершинным* покрытием графа, если у любого ребра графа хотя бы один из концов входит в  $X_v$ . Если считать, что вершина «покрывает» инцидентные ей рёбра, то вершинное покрытие графа  $G$  — это множество вершин, которые покрывают все его рёбра. Размером вершинного покрытия называется число входящих в него вершин, т. е.  $|X_v|$  мощность множества  $X_v$ . Решение задачи о вершинном покрытии подразумевает определение минимально возможного размера вершинного покрытия для заданного графа [1].

В [2] задача размещения сформулирована следующим образом: дано множество элементов  $V = \{e_i, i = \overline{1, n}\}$ , соединённых друг с другом множеством связей  $U \subseteq V \times V$ . В качестве модели данных используется неориентированный граф  $G = (V, U)$ , информация о связях записывается в матрицу смежности  $R = \|r_{ij}\|$ , где  $i, j = \overline{1, n}$ . Под областью размещения понимается множество позиций для установки элементов, организованных в прямоугольную матрицу, имеющих одинаковые линейные размеры и характеризующихся координатами, позиции, объединённые в строки и столбцы, имеют одинаковые  $X$ - и  $Y$ -координаты соответственно. Задача сводится к определению размещения  $P = (p_1, p_2, \dots, p_n)$ , где каждой позиции области размещения соответствует некоторый элемент  $e_i \in V$ , при этом:

$$F = \sum_{i=1}^n \sum_{j=1}^n r_{i,j} \cdot d_{i,j} \rightarrow \min, \quad (1)$$

\* Работа выполнена при финансовой поддержке РФФИ (проект 12-01-00474).

где  $F$  — суммарная длина связей элементов;  $r_{ij}$  — вес ребра в матрице смежности  $R = \|r_{ij}\|$ ;  $d_{ij}$  — длина связи элементов  $e_i$  и  $e_j$ .

В процессе разработки изделия проектировщики ограничены во времени, а также требованиями к качеству изделия, указанными в техническом задании. Подходы, гарантирующие получение оптимальных решений, на практике используются достаточно редко ввиду временной сложности. Используемые на практике стандартные алгоритмы решения проектных задач не гарантируют получение глобального оптимума. Работа данных алгоритмов завершается либо после попадания в локальный оптимум, либо после выполнения заданного числа итераций. Относительно новым алгоритмическим инструментом решения проектных задач являются идеи, основанные на моделировании эволюционных процессов, например, методы моделирования отжига, генетические алгоритмы, автоматы адаптации. Недостатком методов моделирования отжига является зависимость итогового решения от начального, генетических алгоритмов — большой объём вычислений на каждой итерации, а автоматов адаптации — трудности, связанные с представлением исходной задачи в виде адаптивной системы. Несмотря на перечисленные недостатки, данные подходы обладают существенным достоинством — возможностью получать субоптимальные решения за небольшое, как правило, полиномиальное, время. При этом существует априорно неизвестная вероятность улучшения полученного ранее субоптимального решения [1, 3—7]. По этой причине проектирование СБИС, как правило, реализуется на основе итерационных процедур с периодическим повторением реализованных ранее этапов. В ходе выполнения очередной итерации в проект СБИС вносятся локальные изменения, способствующие, по мнению экспертов, улучшению разрабатываемого изделия. Сущность изменений в ряде случаев может состоять в дополнительной локальной оптимизации решённых ранее проектных задач, например, покрытия, разбиения, трассировки, компакций и т. д. как для всей проектируемой схемы, так и для некоторой её части. Таким образом, на практике проектирование элементов СБИС сводится не к нахождению глобальных оптимумов, а к поиску некоторых компромиссных, приемлемых, по мнению разработчиков, решений.

При проектировании особую роль играет опыт эксперта-проектировщика, его способность выбрать наиболее перспективные направления, в соответствии с которыми будет осуществляться дальнейшая доводка изделия, а также способность к анализу получаемых промежуточных решений [1, 3, 7]. В общем случае процессу доводки изделия присуща свойственная неопределённость, связанная с неоднозначностью перспектив локальной оптимизации. Поэтому необходимо разрабатывать системы поддержки доводки проектных решений, взаимодействующие с проектировщиком на основе нечётких инструкций.

**Предлагаемый подход.** Проблема дефицита проектного времени, а также противоречивость перспектив локальной оптимизации (и, как следствие, общей доводки изделия) частично может быть решена за счёт использования нечётких алгоритмов поддержки управления процессом проектирования СБИС. В связи с этим разработана система поддержки процесса доводки СБИС на основе трёх нечётких команд: «незначительно улучшить решение», «улучшить решение», «существенно улучшить решение». Под заданными управляющими инструкциями понимаются не только качественные ожидания величины улучшения исходного решения, но и временные лимиты, необходимые для выполнения соответствующих действий. Будем исходить из того, что ожидаемая величина улучшения исходного решения в некотором роде прямо пропорциональна времени, отводимому для улучшения данного решения. Величины  $|X_i|$  и  $F$  являются достаточно условными характеристиками оптимальности покрытия и размещения, но по причине отсутствия критериев оптимальности, не связанных с полным перебором, они будут использоваться как критерии успешности локальных оптимизаций.

Определим лингвистические переменные «время работы» и «качество решения», принимающие три значения: «малое», «среднее», «большое». Переменная «время работы» определяет длительность решения задачи оптимизации, а «качество решения» — улучшение исходного решения проектной задачи. Введём чёткую градацию переменных «время работы» и «качество решения». В двумерной декартовой системе координат, по аналогии с [8], ассоциируем значения переменной «время работы»  $T$  — с осью абсцисс, а «качество решения»  $Q$  — с осью ординат. Величины  $x_1$ — $x_4$  носят относительный характер и принимают значения от 0 до 1. В общем случае одноимённые, а тем более разноимённые значения различных переменных не равны между собой, например, значение «малое» логической переменной «время работы» не равно значению «малое» логической переменной «качество решения». По этой причине введём постфиксы « $-t$ » и « $-q$ » для идентификации различия значений «малое», «среднее», «большое» переменных «время работы» ( $-t$ ) и «качество решения» ( $-q$ ) (рис. 1).

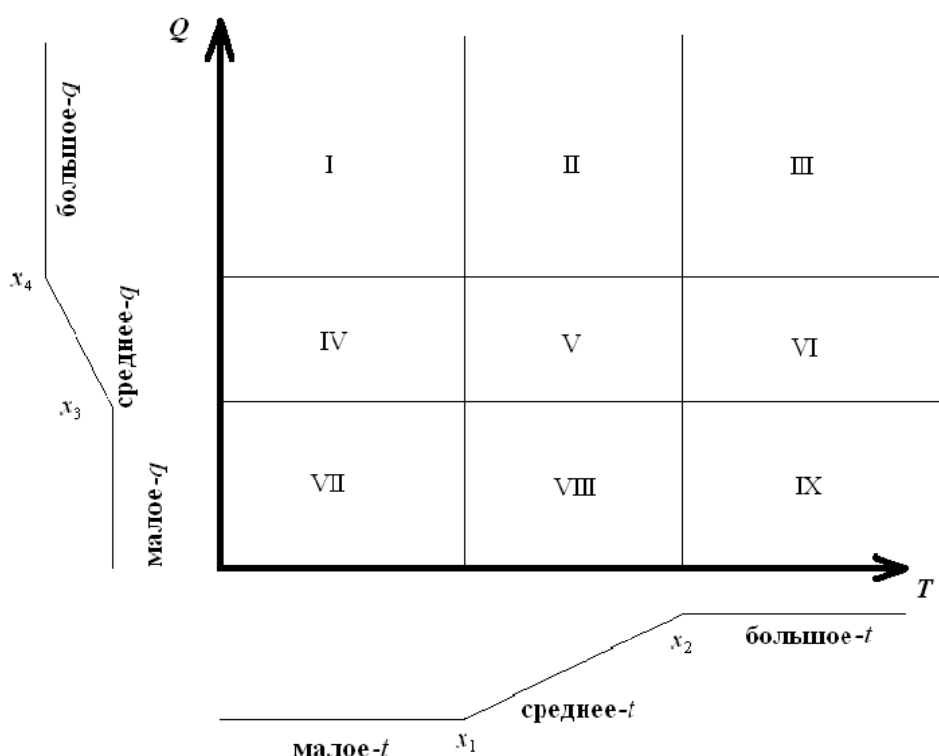


Рис. 1. Чёткое сегментирование области значений

Абсолютная градация значений лингвистических переменных определяется произведением величин  $t_1$  на  $x_1$  или  $x_2$  и  $q_1$  на  $x_3$  или  $x_4$  соответственно, где  $t_1$  — время первой оптимизации проектной задачи,  $q_1$  — количественная оценка первой оптимизации проектной задачи.

При вводе любой из трёх команд («незначительно улучшить решение», «улучшить решение», «существенно улучшить решение») инициализируются два счётчика  $t$  и  $q$ . Счётчик  $t$  фиксирует время работы алгоритма оптимизации,  $q$  — улучшение качества решения. Таким образом, каждая выполняемая или завершённая команда характеризуется кортежем  $(t', q')$ , где  $t'$  — длительность выполнения команды,  $q'$  — полученное улучшение решения. Порядок реализации одной итерации алгоритма улучшения приведён на рис. 2.

Блок «Формирование нечёткой команды» предназначен для выбора нечёткой команды, загрузки условий задачи оптимизации и сегментов области значений нечётких переменных. Блок «Решение задачи» включает в себя операции инициализации, запуск счётчиков  $t$  и  $q$ , решение

задачи оптимизации с учётом поступившей нечёткой команды, а также фиксацию кортежа  $(t', q')$  после окончания решения оптимизационной задачи. Критериями необходимости остановки процесса выполнения команды могут быть: превышение лимита времени, отведённого на выполнение команды; переход в другую качественную или временную градацию  $t'$  и (или)  $q'$ ; выполнение алгоритмом оптимизации заданного числа итераций; не улучшение решения в течение заданного числа итераций. После завершения выполнения лингвистической команды в случае необходимости производится корректировка значений  $x_1$ — $x_4$ , (блок «Анализ результатов»). В связи с тем, что алгоритмы, улучшающие исходные решения, носят вероятностный характер, использование жёстких методов корректировки градаций переменных «время работы» и «качество решения» нецелесообразно. Предлагается использовать адаптивный подход к управлению процессом корректировки градаций значений лингвистических переменных. Для управления процессом корректировки градаций лингвистических команд разработан автомат адаптации (АА), схема которого приведена на рис. 3 [1, 9].

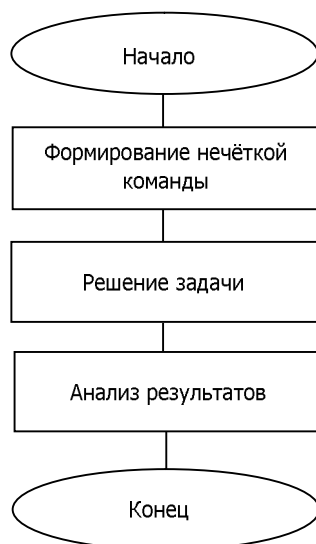


Рис. 2. Схема работы алгоритма интеллектуальной поддержки

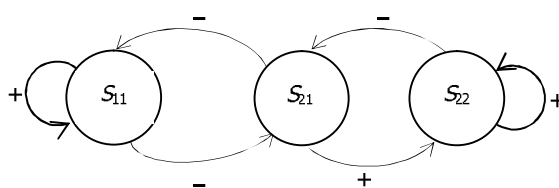


Рис. 3. Схема переходов АА

Частично схема выработки управляющих сигналов поощрения (+) и наказания (–) АА при реализации команды «незначительно улучшить решение» представлена в таблице 1.

Таблица 1

#### Правила выработки сигналов при чёткой системе градаций

| $t'$       | $q'$             | Сегмент | Тек. альтернатива | Сигнал | Рек. альтернатива |
|------------|------------------|---------|-------------------|--------|-------------------|
| $t' < x_1$ | $q' < x_3$       | VII     | A1                | –      | A2                |
| $t' < x_1$ | $q' < x_3$       | VII     | A2                | +      | A2                |
| $t' < x_1$ | $x_3 < q' < x_4$ | IV      | A1                | +      | A1                |
| $t' < x_1$ | $x_3 < q' < x_4$ | IV      | A2                | –      | A1                |

Предлагаемый АА поддерживает две альтернативы  $A_1$  и  $A_2$ . Состояние  $S_{11}$  соответствует альтернативе  $A_1$ , а состояния  $S_{21}$  и  $S_{22}$  — альтернативе  $A_2$ . Альтернатива  $A_1$  заключается в необходимости корректировать градации значений лингвистических переменных, а альтернатива  $A_2$  подразумевает неизменность градаций. Приведённые в таблице данные частично описывают правила выработки управляющих сигналов после реализации команды «незначительно улучшить решение».

Рассмотрим более подробно правила корректировки для некоторых сегментов области значений переменных «время работы» и «качество решения» (рис. 1) на примере выполнения команды «незначительно улучшить решение». В таблице 2 приведены предлагаемые корректирующие воздействия.

Таблица 2

Рекомендуемые корректирующие воздействия

| № сегмента | Условия попадания в сегмент                                                                                                                               | Корректировка                              |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| I          | За малое время получено резкое, существенное улучшение оптимизируемого решения, по этой причине работа алгоритма оптимизации остановлена.                 | $\downarrow(x_1, x_2); \uparrow(x_3, x_4)$ |
| II         | За среднее время получено резкое, существенное улучшение решения. Работа алгоритма остановлена по причине превышения временного или качественного лимита. | $\downarrow(x_1, x_2); \uparrow(x_3, x_4)$ |
| IV         | За малое время получено существенное улучшение решения.                                                                                                   | $\downarrow(x_1, x_2); \uparrow(x_3, x_4)$ |
| VII        | Работа алгоритма остановлена в связи с малым улучшением за приемлемое время, либо по причине отсутствия улучшений в течение нескольких итераций.          | В соответствии с формулой (2)              |

В случае если  $(t', q')$  принадлежит VII сегменту, корректировка  $x_1$  осуществляется по правилу:

$$x_1 := x_1 + [(x_1 - \Delta_1) - (x_1 - t')] \cdot \alpha_1, \quad (2)$$

где  $\Delta_1 \approx 0,85x_1$ ;  $\alpha_1 \approx 0,3$ . Величина  $\Delta_1$  вероятностно определяет направленность корректировки (на снижение или увеличение  $x_1$ ), а  $\alpha_1$  — линейно величину корректирующего воздействия. Значение  $x_2$  изменяется на величину  $(x_1 - \Delta_1) - (x_1 - t')$ . Значения  $x_3$  и  $x_4$  — соответственно по аналогии со значениями  $x_1$  и  $x_2$ . Из формулы (3) следует, что если  $t' > \Delta_1$ , то необходимо увеличить  $x_1$ , в противном случае ( $t' < \Delta_1$ ) — уменьшить. Аналогичные правила разработаны для различных соотношений элементов кортежа  $(t', q')$  со значениями  $x_1$ – $x_4$ .

Чёткое разбиение области значений подразумевает некоторую скачкообразность формирования качественных оценок. По этой причине предложена расплывчатая градация значений нечётких переменных «время работы» и «качество решения» (рис. 4).

Для каждого элемента области рассуждений переменные «время работы» и «качество решения» принимают значения «малое», «среднее», «большое» с определённой степенью достоверности. Нижней границей степени достоверности является «0», верхней — «1». Соответствие элемента области рассуждений значениям переменной «малое», «среднее», «большое» определяется на основе функций принадлежности, аргументом которых является рассматриваемый элемент. Функции принадлежности задаются трапециями, описываемыми кортежами  $(a_p, b_p, c_p, d_p)$ . В рассматриваемом случае  $a_p, b_p, c_p, d_p$  — не отрицательные числа, подстрочный индекс  $p$  — признак принадлежности элементов кортежа к трапеции, описывающей одно из значений лингвистических переменных «время работы» или «качество решения». Величина  $\mu_p(x)$  определяет степень принадлежности элемента  $x$  к значению переменной, соответствующей кортежу  $(a_p, b_p, c_p, d_p)$ . Значение величины  $\mu_p(x)$  определяется по формуле:

$$\mu_p(x) = \begin{cases} \frac{x - a_p}{b_p - a_p}, & \text{если } a_p \leq x \leq b_p, \\ 1, & \text{если } b_p \leq x \leq c_p, \\ \frac{d_p - x}{d_p - c_p}, & \text{если } c_p \leq x \leq d_p, \\ 0, & \text{в остальных случаях.} \end{cases} \quad (3)$$

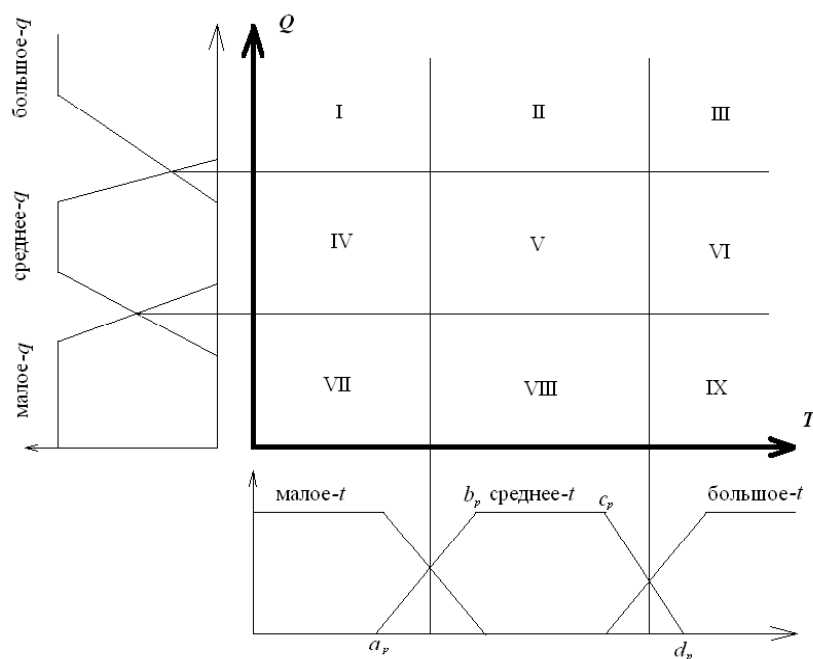


Рис. 4. Расплывчатое сегментирование области значений

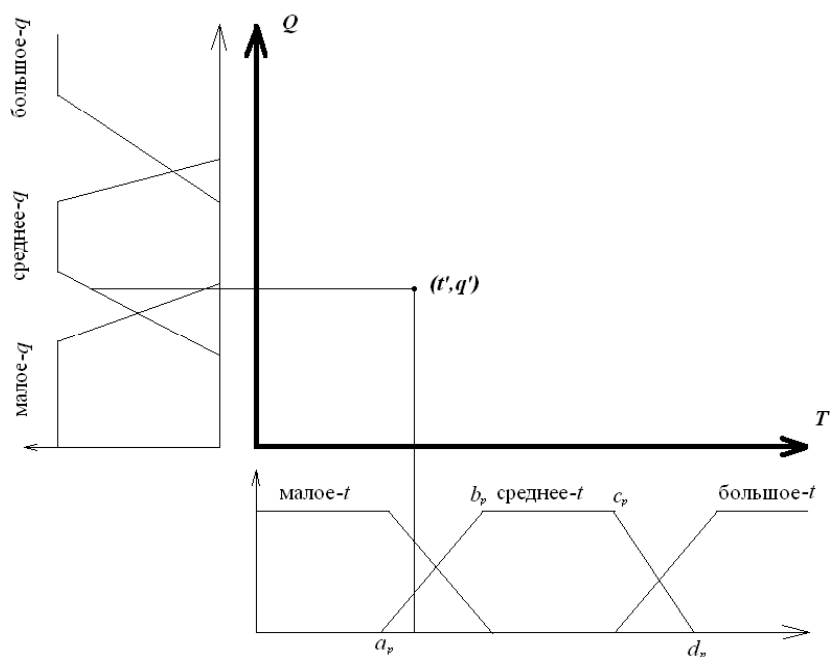


Рис. 5. Пример нечёткой принадлежности кортежа  $(t', q')$

При расплывчатой сегментации области значений элементам каждого кортежа  $(t', q')$  соответствуют степени принадлежности значениям «малое», «среднее», «большое» лингвистических переменных «время работы» и «качество решения». В общем случае при такой градации, наряду с чёткими областями, однозначно и абсолютно соответствующими значениям лингвистических переменных, присутствуют нечёткие — частично соответствующие значениям лингвистических переменных. При этом элементу кортежа  $(t', q')$  могут соответствовать несколько ненулевых значений функции принадлежности к нечёткой переменной, например, на рис. 5 элементу  $t'$  соответствуют ненулевые степени принадлежности значениям «малое- $t$ » и «среднее- $t$ » лингвистической переменной «время решения».

Для кортежа  $(t', q')$ , представленного на рис. 5,  $\mu_{\text{«малое-}t\text{»}}(t') \approx 0,75$ ,  $\mu_{\text{«среднее-}t\text{»}}(t') \approx 0,32$ ,  $\mu_{\text{«малое-}q\text{»}}(q') \approx 0,09$ , а  $\mu_{\text{«среднее-}q\text{»}}(q') \approx 0,8$ . Изображённую на рис. 5 ситуацию можно трактовать двойственно — в зависимости от поступившей команды:

— если была подана команда «незначительно улучшить решение», то, поскольку за время, близкое к малому, получено близкое к среднему улучшение, работу алгоритма оптимизации можно завершить;

— если была подана команда «улучшить решение», то, поскольку за время, близкое к малому, получено близкое к среднему улучшение, работу алгоритма оптимизации можно продолжить до получения (по возможности) среднего улучшения за среднее время.

Определим понятие доминирующего значения лингвистической переменной как значения с наибольшей степенью принадлежности. Тогда  $TMax(\mu_{*t}(t'))$  — доминирующее значение нечёткой переменной «время работы», ассоциируемое со значением  $t'$ ;  $QMax(\mu_{*q}(q'))$  — доминирующее значение нечёткой переменной «качество решения», ассоциируемое со значением  $q'$ . Для кортежа  $(t', q')$ , изображённого на рис. 5,  $TMax(\mu_{*t}(t')) = \text{«малое-}t\text{»}$ ,  $QMax(\mu_{*q}(q')) = \text{«среднее-}q\text{»}$ .

Степень доминирования — насколько степень принадлежности доминирующего значения логической переменной выше степени принадлежности оставшихся значений. Для примера, изображённого на рис. 5, доминирующими значениями лингвистических переменных «время работы» и «качество решения» являются «малое- $t$ » и «среднее- $q$ » соответственно. Численные характеристики доминирования значений «малое- $t$ » и «среднее- $q$ » определим соответственно по формулам:

$$D(t') = \mu_{\text{«малое-}t\text{»}}(t') - \mu_{\text{«среднее-}t\text{»}}(t') \approx 0,75 - 0,32 \approx 0,43;$$

$$D(q') = \mu_{\text{«среднее-}q\text{»}}(q') - \mu_{\text{«малое-}q\text{»}}(q') \approx 0,8 - 0,09 \approx 0,71.$$

При нечёткой системе градаций выработку управляющих сигналов целесообразно осуществлять с учётом значения величины  $D$ . При нечёткой системе градаций используется схема АА, изображённого на рис. 3, с модифицированными правилами генерации управляющих сигналов, которые приведены в таблице 3.

Таблица 3

Правила выработки сигналов при нечёткой системе градаций

| Сегмент | Текущая альтернатива | Сигнал                  | Рекомендуемая альтернатива |
|---------|----------------------|-------------------------|----------------------------|
| IV      | A1                   | +<br>с вероятностью $D$ | A1                         |
| IV      | A2                   | -<br>с вероятностью $D$ | A1                         |

**Выводы.** Разработан алгоритм интеллектуальной поддержки улучшения промежуточных решений оптимизационных задач. Особенностью алгоритма является адаптивная реализация лингвистических команд на основе чёткой и расплывчатой системы градаций значений лингвистических переменных. Чёткая система градаций более проста с точки зрения реализации, расплывчатая — более чувствительна к динамике вычислительного процесса. При расплывчатой градации наличие нескольких ненулевых значений функции принадлежности у элементов области значений позволяет организовывать вероятностно-направленное управление поиском проектных решений на основе анализа доминирующих значений лингвистических переменных.

#### **Библиографический список**

1. Курейчик, В. М. Поисковая адаптация: теория и практика / В. М. Курейчик, Б. К. Лебедев, О. Б. Лебедев // Монография. Москва, ФИЗМАТЛИТ, 2006. — 272 с.
2. Бакало, М. А. Модифицированный алгоритм размещения методом парных перестановок / М. А. Бакало, В. В. Курейчик // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». — Таганрог: Изд-во ТРТУ, 2007. — № 1 (73). — С. 77—84.
3. Морозов, К. К. Автоматизированное проектирование конструкции радиоэлектронной аппаратуры / К. К. Морозов, В. Г. Одинокоев, В. М. Курейчик // Учеб. пособие для вузов. — Москва, Радио и связь, 1983. — 280 с.
4. Donath, W. E. Timing Driven Placement Using Complete Path Delays Proc / W. E. Donath // ACM/IEEE DAC, 2008.
5. Курейчик, В. М. Биоинспирированный поиск с использованием сценарного подхода / В. М. Курейчик // Известия ЮФУ. Технические науки. — 2010. — № 7 (108). — С. 7—12.
6. Cong, J. MARS — A multilevel full-chip gridless routing system / J. Cong, J. Fang, M. Xie, Y. Zhang // IEEE Trans Comput.-Aided Design Integr. Syst. — 2005. — Vol. 24, № 3. — P. 382—394.
7. Лебедев, Б. К. Поисковые процедуры канальной трассировки, базирующиеся на моделировании адаптивного поведения роя частиц в пространстве решений с неупорядоченным лингвистическим шкалированием / Б. К. Лебедев, В. Б. Лебедев // Известия ЮФУ. Технические науки. — 2009. — № 12 (101). — С. 15—22.
8. Banks, Walter. Fuzzy logic in embedded microcomputers and control systems / Walter Banks, Gordon Hayward // Published by Byte Craft Limited, Waterloo Ontario Canada, 2001.
9. Чернышёв, Ю. О. Адаптивная оптимизация запросов в современных системах управления базами данных / Ю. О. Чернышёв, Н. Н. Венцов, А. В. Крупенин, А. В. Щербаков, С. А. Мухтаров // Монография. Краснодар: ФВАС, 2011. — 89 с.

Материал поступил в редакцию 14.06.2012.

#### **References**

1. Kurejchik, V. M. Poiskovaya adaptaciya: teoriya i praktika / V. M. Kurejchik, B. K. Lebedev, O. B. Lebedev // Monografiya. Moskva, FIZMATLIT, 2006. — 272 s. — In Russian.
2. Bakalo, M. A. Modificirovannyj algoritm razmeshheniya metodom parnyx perestanovok / M. A. Bakalo, V. V. Kurejchik // Izvestiya TRTU. Tematicheskij vy`pusk «Intellectual`ny`e SAPR». — Taganrog: Izd-vo TRTU, 2007. — № 1 (73). — S. 77—84. — In Russian.
3. Morozov, K. K. Avtomatizirovannoe proektirovanie konstrukcii radioe`lektronnoj apparatury` / K. K. Morozov, V. G. Odinokov, V. M. Kurejchik // Ucheb. posobie dlya vuzov. — Moskva, Radio i svyaz`, 1983. — 280 s. — In Russian.
4. Donath, W. E. Timing Driven Placement Using Complete Path Delays Proc / W. E. Donath // ACM/IEEE DAC, 2008.

5. Kurejchik, V. M. Bioinspirovanny`j poisk s ispol`zovaniem scenarnogo podxoda / V. M. Kurejchik // Izvestiya YuFU. Texnicheskie nauki. — 2010. — № 7 (108). — S. 7—12. — In Russian.
6. Cong, J. MARS — A multilevel full-chip gridless routing system / J. Cong, J. Fang, M. Xie, Y. Zhang // IEEE Trans Comput.-Aided Design Integr. Syst. — 2005. — Vol. 24, № 3. — P. 382—394.
7. Lebedev, B. K. Poiskovy`e procedury` kanal`noj trassirovki, baziruyushhiesya na modelirovanii adaptivnogo povedeniya roya chasticz v prostranstve reshenij s neuporyadochenny`m lingvisticheskim shkalirovaniem / B. K. Lebedev, V. B. Lebedev // Izvestiya YuFU. Texnicheskie nauki. — 2009. — № 12 (101). — S. 15—22. — In Russian.
8. Banks, Walter. Fuzzy logic in embedded microcomputers and control systems / Walter Banks, Gordon Hayward // Published by Byte Craft Limited, Waterloo Ontario Canada, 2001.
9. Cherny`shyov, Yu. O. Adaptivnaya optimizaciya zaprosov v sovremenny`x sistemax upravleniya bazami danny`x / Yu. O. Cherny`shyov, N. N. Venczov, A. V. Krupenin, A. V. Shherbakov, S. A. Muxtarov // Monografiya. Krasnodar: FVAS, 2011. — 89 s. — In Russian.

## **ALGORITHM DESIGN OF INTELLECTUAL SUPPORT FOR PERFECTING OPTIMIZATION PROBLEM INTERMEDIATE SOLUTIONS**

**Y. O. Chernyshev, N. N. Ventsov**

(Don State Technical University),

**S. A. Mukhtarov**

(Military Academy of Communication, Krasnodar Branch)

*The problem areas requiring the intellectual support for solving optimization tasks are considered. The adaptive algorithm of the optimization tasks solution support on the basis of the linguistic commands is developed. Both accurate and fuzzy parameter characteristics of the calculating process could be associated to the commands.*

**Keywords:** adaptation, intellectual systems, optimization.