

4. Potts C.I., Giddens T.D., Yadav S.B. The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial selection. IEEE Trans. on Systems, Man and Cybernetics, vol.24, No.1, Summary 1994.
5. Гладков Л.А. Методы генетического поиска: Монография / Л.А. Гладков, Л.А. Зинченко, В.В. Курейчик. – Таганрог: Изд-во ТРТУ, 2002.
6. Букатова И.Л. Эволюционное моделирование и его приложения./И.Л. Букатова – М.: Наука, 1994.
7. Штовба С.Д. Муравьиные алгоритмы. Exponenta Pro. Математика в приложениях, 2003, №4. – С. 70-75.
8. Kirkpatrick S., Gelatt C. D., Vecchi M. P. Optimization by simulated annealing. Science. v220 (1983), pp. 671-680.
9. Binder K. Monte Carlo methods in statistical physics. Berlin: Springer, 1978.
10. Aarts E. H. L., Korst J. H. M., Laarhoven van P. J. M. Simulated annealing. in Aarts E., Lenstra J.K. (Eds) Local search in combinatorial optimization. Chichester: Wiley, 1997. pp. 91-120.
11. Aarts E. H. L., Korst J. H. M. Simulated annealing and Boltzmann machines. Chichester: Wiley, 1989.
12. Hajek B., Sasaki G. Simulated annealing: to cool it or not. Sys. Contr. Lett. v12 (1989), pp. 443-447.

УДК 519.712.2

Л.А. Гладков, Н.В. Гладкова

РЕШЕНИЕ ЗАДАЧИ КОММИВОЯЖЕРА НА ОСНОВЕ НЕЧЕТКОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА*

Нечеткий генетический алгоритм можно определить как алгоритм, сочетающий поисковые возможности генетических алгоритмов и возможности математического аппарата нечеткой логики. Математический аппарат теории нечетких систем используется в данном случае для кодирования, подбора оптимальных параметров генетических алгоритмов, значений вероятности генетических операторов, выбора функции пригодности и критерия останова, создания нечетких генетических операторов.

Одной из особенностей нечеткого генетического алгоритма является наличие нечеткого логического контроллера (НЛК). НЛК преобразует заданные параметры к нечеткому виду, затем с помощью Базы Правил (БП) определяет управляющее воздействие и возвращает скорректированные значения контрольных параметров.

В составе НЛК можно выделить следующие блоки [1]:

- ◆ база знаний, включающая в себя базу правил и базу данных;
- ◆ блок фаззификации;
- ◆ блок дефаззификации;
- ◆ система вывода решения;
- ◆ система контроля.

База правил, иногда называемая лингвистической моделью, представляет собой множество нечетких правил $R^{(k)}$, $k = 1, \dots, N$. НЛК оперирует нечеткими множествами. Поэтому конкретное значение входного параметра НЛК подлежит операции *фаззификации*, в результате которой ему будет сопоставлено нечеткое множество.

На выходе блока выработки решения формируется одно или несколько нечетких множеств с соответствующими функциями принадлежности. Встает задача

* Работа выполнена при поддержке РФФИ (грант № 08-01-00473).

преобразования этих результирующих нечетких множеств (нечеткого множества) в управляющее воздействие на объекты управления НЛК. Такое преобразование называется *дефаззификацией (defuzzification)*.

Очевидно, что эффективность работы НГА зависит от качества разработанных блоков и степени их адекватности решаемой задаче.

В генетических алгоритмах (ГА) управляющими параметрами, как правило, являются значения вероятности выполнения генетических операторов кроссинговера (P_c) и мутации (P_m), а также размер популяции. В нечетком генетическом алгоритме происходит динамическое изменение этих параметров при помощи нечеткого логического контроллера (НЛК). Нечеткий логический контроллер на основе входных параметров рассчитывает управляющее воздействие, регулирующее параметры ГА.

Постановку задачи о коммивояжере можно описать следующим образом: Есть некий коммивояжер, которому необходимо посетить N городов, не заезжая в один и тот же город дважды, и вернуться в исходный пункт. При этом длина (стоимость) пройденного пути должна быть минимальной. С точки зрения теории графов постановка задачи выглядит так: дан граф $G = (X, U)$, где $|X| = n$ – множество вершин (городов), $|U| = m$ – множество ребер (путей) между городами; $D_{n \times n}$ – матрица расстояний (стоимостей), где d_{ij} , $i, j \in \{1, 2, \dots, n\}$ – это длина пути (стоимость переезда) из города (вершины) x_i в x_j .

Для решения задачи коммивояжера предлагается использовать нечеткий генетический алгоритм. При этом мы используем следующие настройки ГА:

1. Случайное формирование начальной популяции, например, на основе метода «дробовика».
2. Кодирование решений в виде последовательности проходимых путей (ребер).
3. Оператор кроссинговера на основе жадной эвристики.
4. Оператор мутации.
5. Оператор отбора на основе «колеса рулетки».

Основным моментом алгоритма является применение НЛК, использующего в качестве входных параметров среднее и лучшее значение ЦФ.

Нечеткий логический контроллер оперирует лингвистическими переменными, такими как: NL (negative large) – «значительно хуже», NS (negative small) – «незначительно хуже», ZE (zero) – «частичное изменение», PS (positive small) – «незначительно лучше», PL (positive large) – «значительно лучше» [2, 3].

Формирование управляющего воздействия НЛК осуществляет на основе оценки двух параметров [4, 5]:

$$e_1(t) = \frac{f_{ave}(t) - f_{best}(t)}{f_{ave}(t)},$$

$$e_2(t) = \frac{f_{ave}(t) - f_{ave}(t-1)}{f_{best}(t)},$$

где t – временной шаг; $f_{max}(t)$ – лучшее значение ЦФ на итерации t ; $f_{ave}(t)$ – среднее значение ЦФ на итерации t ; $f_{ave}(t-1)$ – среднее значение ЦФ на итерации $(t-1)$.

Параметры e_1 , e_2 заданы на следующих интервалах: $e_1 \in [0; 1]$; $e_2 \in [-1; 1]$.

Выходной параметр $\Delta Pc(t)$ определяющий вероятность выполнения кроссинговера задан на интервале: $\Delta Pc(t) \in [-0.1; 0.1]$. Выходной параметр для оператора мутации $\Delta Pm(t)$ тот, же что и для кроссинговера. Отсюда видно, что вероятности могут меняться не более, чем на 10%. Тогда используем четкие значения, чтобы изменить вероятности Pc и Pm следующим образом:

$$Pc(t) = Pc(t-1) + \Delta Pc(t);$$

$$Pm(t) = Pm(t-1) + \Delta Pm(t).$$

То есть, НЛК представляет собой двумерный набор нечетких правил, представленных в табл. 1, 2.

Таблица 1

Нечеткие правила для ОК ($\Delta Pc(t)$)

	e_2				
e_1	NL	NS	ZE	PS	PL
PL	NS	ZE	PS	PS	PL
PS	ZE	ZE	ZE	ZE	ZE
ZE	NS	NL	ZE	NL	NS

Таблица 2

Нечеткие правила для ОМ ($\Delta Pm(t)$)

	e_2				
e_1	NL	NS	ZE	PS	PL
PL	PS	ZE	PS	NS	NL
PS	ZE	ZE	NS	ZE	NS
ZE	PS	PL	ZE	PL	PS

Так, например, если на входе нечеткого логического контроллера заданы следующие параметры $e_1 = ZE$ и $e_2 = NL$, то, согласно приведенным таблицам (см. табл. 1, 2), получим:

$$\Delta Pc(t) = NS, \Delta Pm(t) = PS.$$

Следовательно, значения вероятностей кроссинговера и мутации определяются выражениями:

$$Pc(t) = Pc(t-1) + NS;$$

$$Pm(t) = Pm(t-1) + PS.$$

То есть, при заданных значениях параметров e_1 и e_2 вероятность кроссинговера «немного» уменьшится, а вероятность мутации «значительно» увеличится.

Очевидно, что для эффективной работы нечеткого генетического алгоритма необходимо провести тестирование и настройку, как входных сигналов, так и соответствующих им управляющих воздействий нечеткого логического контроллера [6].

Структура предложенного нечеткого генетического алгоритма выглядит следующим образом (рис. 1):

```

Начало НГА
t = 0 Iteration counter
Формирование начальной популяции P(t)
Расчет ЦФ популяции P(t)
Пока (не выполнено условие останова)
{
    t = t + l
    Селекция
    ОК P(t)
    ОМ P(t)
    Расчет ЦФ P(t)
    Отбор P(t)
    Регулировка параметров ГА
}
    
```

```

{
  НЛК ( $e_1, e_2$ )
  Обновляем параметры
}
}
Конец НГА

```

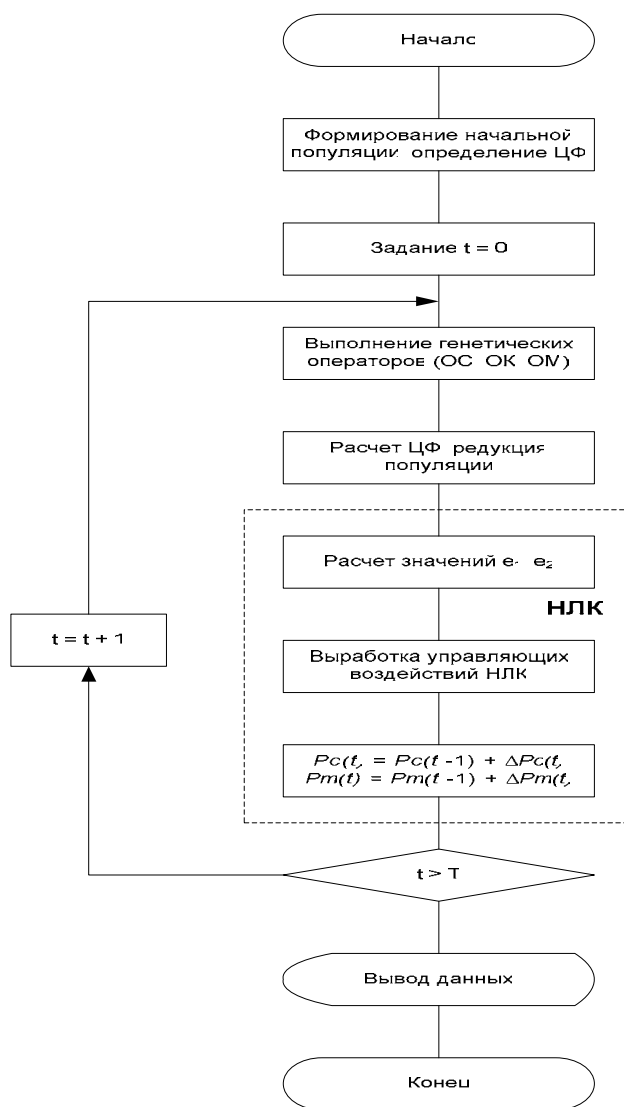


Рис. 1. Структурная схема НГА

На основе предложенного подхода была разработана программа для ЭВМ, в которой реализован описанный выше НГА. Программа создана в среде разработки Borland C++ Builder 6.0. Внешний вид программы представлен на рис. 2.

Кроме задания основных параметров ГА, в программе имеется возможность выбора варианта решения задачи – с использованием НЛК и без него. Тем самым, можно легко проводить сравнительный анализ стандартного и нечеткого ГА.

В программе также реализована возможность задания нечетких правил, что делает удобным процесс исследования и выявления оптимальных сочетаний настроек нечеткого генетического алгоритма (рис. 3).

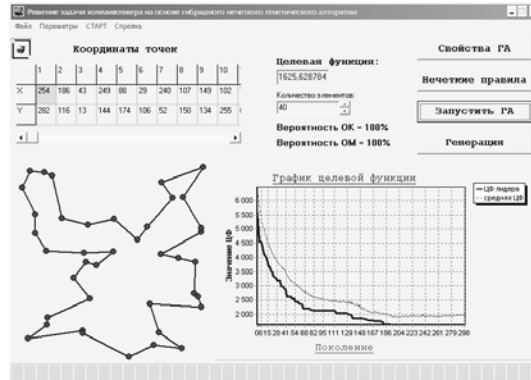


Рис. 2. Основное окно программы

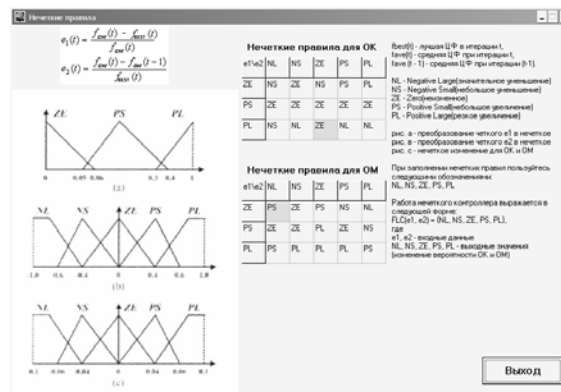


Рис. 3. Задание нечетких правил

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Herrera F., Lozano M. Fuzzy Adaptive Genetic Algorithms: design, taxonomy, and future directions // Soft Computing 7(2003), Springer-Verlag, 2003. – p. 545-562.
2. Hongbo Liu, Zhanguo Xu, Ajith Abraham. Hybrid Fuzzy-Genetic Algorithm Approach for Crew Grouping. – Source unknown.
3. Fayad C., Petrovic S. A Genetic Algorithm for the Real-World Fuzzy Job Shop Scheduling. School of Computer Science and Information Technology University of Nottingham, <http://www.cs.nott.ac.uk/~cxf.~sxp>
4. Ярушкіна Н.Г. Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004.
5. Нечеткие гибридные системы. Теория и практика / Под ред. Н.Г. Ярушкіной. – М.: Физматлит, 2007.
6. Гладков Л.А. Решение задач и оптимизации решений на основе нечетких генетических алгоритмов и многоагентных подходов // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТРТУ, 2006. №8(63). – С. 83-88.