

**О ГЕНЕТИЧЕСКОМ АЛГОРИТМЕ
РЕШЕНИЯ ЗАДАЧИ ВОЗМОЖНОСТНОЙ ОПТИМИЗАЦИИ
ПРИ ВЗАИМНО T_W -СВЯЗАННЫХ ПАРАМЕТРАХ**

И.С. Солдатенко

Кафедра информационных технологий

Поступила в редакцию 29.11.2007, после переработки 04.02.2008.

В работе описывается генетический алгоритм решения задачи возможностной оптимизации в максимаксной постановке при взаимно T_W -связанных параметрах. Осуществляется сравнительный анализ эквивалентных детерминированных аналогов этой задачи в случае минисвязанных параметров и T_W -связанных. Возможности метода демонстрируются на модельном примере, снабженном графическими иллюстрациями.

Genetic algorithm for fuzzy level optimization problem in case of mutually T_W -related parameters is described. Comparative analysis of equivalent crisp analogues for the same optimization problem which has in one case min-related and in another case — T_W -related parameters is given. The article contains numerical example and a series of illustrations for criterion functions' domains defined by restrictions in both cases.

Ключевые слова: возможностное программирование, задача максимаксной (уровневой) оптимизации, взаимно T_W -связанные параметры, генетический алгоритм.

Keywords: possibilistic mathematical programming, level optimization, genetic algorithm, mutually T_W -related parameters.

1. Постановка задачи

В работе [2] была исследована следующая задача возможностного программирования [4, 6, 7] в случае взаимно T_W -связанных параметров:

$$k \rightarrow \max, \quad (1)$$

$$\pi \{f_0(x, \gamma) = k\} \geq \alpha_0, \quad (2)$$

$$\begin{cases} \pi \{f_i(x, \gamma) = 0\} \geq \alpha_i, & i = 1, \dots, m, \\ x \in X. \end{cases} \quad (3)$$

В этой задаче $f_0(x, \gamma) = \sum_{j=1}^n a_{0j}(\gamma)x_j$, $f_i(x, \gamma) = \sum_{j=1}^n a_{ij}(\gamma)x_j - b_i(\gamma)$, $i = 1, \dots, m$. Предполагается, что $a_{0j}(\gamma)$, $a_{ij}(\gamma)$ и $b_i(\gamma)$ — взаимно T_W -связанные [5] нечеткие величины (L, R)-типа [1]:

$$a_{0j}(\gamma) = (a'_{0j}, a''_{0j}, \eta_{0j}, \beta_{0j})_{LR}, \quad a_{ij}(\gamma) = (a'_{ij}, a''_{ij}, \eta_{ij}, \beta_{ij})_{LR},$$

$$b_i(\gamma) = (b'_i, b''_i, \eta_i, \beta_i)_{LR}, x \in \mathbb{E}_+^n.$$

Эта задача известна как задача уровней, или максимаксной возможностной оптимизации [6, 7]. Для случая минисвязанных параметров она достаточно полно исследована в работах [6, 7, 8].

Для случая взаимно T_W -связанных параметров и идентичных функций представления формы, моделирующих распределения нечетких параметров, в [2] был построен эквивалентный детерминированный аналог (1)-(3) (при условии существования обратных функций представления форм L^{-1} и R^{-1}):

$$k \rightarrow \max, \quad (4)$$

$$\begin{cases} \sum_{j=1}^n a'_{0j} x_j - \max_{j=1, \dots, n} \{x_j \eta_{0j}\} L^{-1}(\alpha_0) \leq k, \\ \sum_{j=1}^n a''_{0j} x_j + \max_{j=1, \dots, n} \{x_j \beta_{0j}\} R^{-1}(\alpha_0) \geq k, \end{cases} \quad (5)$$

$$\begin{cases} \sum_{j=1}^n a'_{ij} x_j - \max_{j=1, \dots, n} \{x_j \eta_{ij}\} L^{-1}(\alpha_i) \leq b''_i + \beta_i R^{-1}(\alpha_i), & i = 1, \dots, m, \\ \sum_{j=1}^n a''_{ij} x_j + \max_{j=1, \dots, n} \{x_j \beta_{ij}\} R^{-1}(\alpha_i) \geq b'_i - \eta_i L^{-1}(\alpha_i), & i = 1, \dots, m, \\ x \in X, \end{cases} \quad (6)$$

где (4)-(5) есть эквивалентный детерминированный аналог модели критерия, а (6) — эквивалентный детерминированный аналог модели ограничений.

Упростим детерминированный аналог модели критерия (4)-(5). Обозначим:

$$f'(x) = \sum_{j=1}^n a'_{0j} x_j - \max_{j=1, \dots, n} \{x_j \eta_{0j}\} L^{-1}(\alpha_0),$$

$$f''(x) = \sum_{j=1}^n a''_{0j} x_j + \max_{j=1, \dots, n} \{x_j \beta_{0j}\} R^{-1}(\alpha_0).$$

Тогда модель (4)-(5) можно переписать в виде:

$$\begin{aligned} k &\rightarrow \max, \\ f'(x) &\leq k \leq f''(x). \end{aligned} \quad (7)$$

Модель (7) можно свести к эквивалентной модели, не содержащей дополнительной (уровневой) переменной:

$$\begin{aligned} f''(x) &\rightarrow \max \\ f''(x) - f'(x) &\geq 0. \end{aligned}$$

В результате задача (4)-(6) переписывается в виде:

$$f''(x) \rightarrow \max, \quad (8)$$

$$\left\{ \begin{array}{l} f''(x) - f'(x) \geq 0, \\ \sum_{j=1}^n a'_{ij} x_j - \max_{j=1, \dots, n} \{x_j \eta_{ij}\} L^{-1}(\alpha_i) \leq b'_i + \beta_i R^{-1}(\alpha_i), \quad i = 1, \dots, m, \\ \sum_{j=1}^n a''_{ij} x_j + \max_{j=1, \dots, n} \{x_j \beta_{ij}\} R^{-1}(\alpha_i) \geq b'_i - \eta_i L^{-1}(\alpha_i), \quad i = 1, \dots, m, \\ x \in X. \end{array} \right. \quad (9)$$

Как было показано в [2], задача (8)-(9) является в общем случае задачей многоэкстремальной и невыпуклой. Ввиду этого для решения эквивалентного детерминированного аналога целесообразным является применение генетического алгоритма оптимизации.

2. Спецификация генетического алгоритма оптимизации для решения задачи

Общую схему генетического алгоритма можно представить следующим образом [3]:

1. Инициализация параметров алгоритма, в частности *popsizе*, *a*, P_c и P_m , где:
 - *popsizе* — количество хромосом в каждой популяции;
 - *a* — константа, влияющая на алгоритм отбора хромосом;
 - P_c — вероятность кроссовера (примерно $P_c \times \textit{popsizе}$ хромосом дадут новые хромосомы);
 - P_m — вероятность мутации (примерно $P_m \times \textit{popsizе}$ хромосом мутируют).
2. Инициализация исходной популяции *popsizе* хромосомами.
3. Модификация текущего набора хромосом при помощи операций кроссовера и мутации.
4. Вычисление критериальной функции для каждой хромосомы.
5. Вычисление значения функции пригодности для каждой хромосомы с использованием критериальной функции.
6. Отбор хромосом при помощи алгоритма рулетки и формирование новой популяции.
7. Повторение шагов 3–6 необходимое количество раз.
8. Вывод лучшей хромосомы в качестве оптимального решения задачи.

Так как наилучшая хромосома i -той популяции необязательно перейдет в $(i + 1)$ -ую популяцию, то мы будем хранить значение наилучшей хромосомы, полученной на предыдущих этапах работы алгоритма. Если в дальнейшем мы найдем более хорошую хромосому, то мы запомним ее вместо прежней.

Рассмотрим более подробно все шаги алгоритма.

2.1 Хромосомы

При помощи хромосом в генетических алгоритмах кодируются решения поставленной задачи оптимизации. Согласно [3] можно использовать как минимум два способа представления хромосом — двоичный и вещественный. В первом случае все компоненты вектора неизвестных $x = (x_1, x_2, \dots, x_n)$ переводятся в двоичный вид и конкатенируются друг с другом. После чего все операции кроссовера и мутации производятся над последовательностями нулей и единиц. Во втором случае генетический алгоритм работает непосредственно с самим вектором x без его кодирования в другой вид. Оба метода имеют свои преимущества и недостатки.

Мы остановимся на втором способе, для того чтобы не усложнять алгоритм операциями кодирования и декодирования двоичных строк и иметь наглядность генетического процесса, которая в общем случае отсутствует, если хромосомы кодируются двоичными строками.

2.2 Инициализация параметров алгоритма

Параметры позволяют управлять процессом работы генетического алгоритма и от их выбора зависит в общем случае эффективность его работы. Имея программную реализацию алгоритма, мы можем экспериментально подобрать значения параметров, при которых поставленная задача будет решаться наиболее эффективно. Первоначальный же выбор значений параметров основывается либо на оценках экспертов, знающих предметную область решаемой задачи, либо эти параметры выбираются наугад.

Один из наиболее важных параметров — размер популяции. При слишком маленьком размере популяции шаг алгоритма будет выполняться быстро, однако общая сходимость алгоритма к оптимальному решению будет медленной, к тому же повышается риск застревания в области локального оптимума. При слишком большой популяции выполнение одного шага алгоритма по построению новой популяции будет занимать значительное время, хотя диапазон работы такого алгоритма очень широк и вероятность нахождения глобального экстремума за меньшее число итераций алгоритма выше, чем в первом случае. Но из-за того, что каждая итерация выполняется долго, и сам алгоритм как и в первом случае будет работать долго.

2.3 Инициализация исходной популяции

Процесс инициализации исходной популяции заключается в присвоении всем *popsize* хромосомам в общем случае произвольных значений, которые тем не менее находятся в области допустимых решений поставленной задачи.

2.4 Выполнение операции кроссовера

Для выполнения операции кроссовера мы отбираем $P_c \times \text{popsize}$ хромосом случайным образом. Будем делать это следующим образом. Для всех $i = 1 \dots \text{popsize}$ будем случайным образом генерировать число $r \in [0, 1]$. При этом r должно иметь равномерное распределение на данном отрезке. Если $r < P_c$, то i -тая хромосома отбирается для операции кроссовера.

В результате мы получим набор из примерно $P_c \times \text{popsize}$ хромосом — $V_1, V_2, V_3, \dots, V_c$, $c \approx P_c \times \text{popsize}$. В операции кроссовера должны участвовать пары хромосом, поэтому если в полученном наборе нечетное их количество, то одну можно выбросить. Формируем случайным образом пары хромосом $(V'_1, V'_2), (V'_3, V'_4), \dots$

Рассмотрим пару (V'_1, V'_2) . Сгенерируем случайное число $c \in (0, 1)$. После этого выполним операцию кроссовера для (V'_1, V'_2) :

$$X = c \times V'_1 + (1 - c) \times V'_2 \quad Y = (1 - c) \times V'_1 + c \times V'_2.$$

X и Y — потомки двух исходных хромосом. Если бы область допустимых решений была выпуклой, то данная операция кроссовера гарантировала бы нам, что X и Y тоже принадлежат области допустимых решений. Однако в нашем случае это не так, поэтому после получения X и Y необходимо проверить, не вышли ли они за пределы области. Если одна из новых хромосом вышла из области, то мы сохраняем вторую и выполняем операцию кроссовера еще раз. Так повторяем до тех пор, пока либо не получим две допустимых разных хромосомы, либо пока не превысим допустимое число итераций. Затем исходные хромосомы V'_1 и V'_2 заменяются новыми X и Y (если таковые есть).

2.5 Выполнение операции мутирования

Отбираем $P_m \times \text{popsize}$ хромосом, которые будут подвергаться операции мутирования, случайным образом. Будем делать это так же, как и в п.2.4. Для всех $i = 1 \dots \text{popsize}$ случайным образом генерируем число $r \in [0, 1]$. Если $r < P_m$, то i -тая хромосома отбирается для операции мутирования.

Затем для каждой отобранной хромосомы V_i'' выполняем операцию мутирования следующим образом. Случайным образом выбираем достаточно большое число M (диапазон этого числа определяется также в пункте 1 алгоритма). Затем также случайным образом генерируем вектор направления $d \in \mathbb{R}^n$. Формируем мутированную хромосому по формуле:

$$Z = V_i'' + M \times d.$$

Если Z вышла за область допустимых решений задачи, то выбираем случайным образом число $M' \in (0, M)$ и повторяем операцию снова. Так делаем до тех пор, пока либо не получим новую допустимую хромосому, либо пока не превысим заданное число итераций операции мутирования. Затем заменяем исходную хромосому V_i'' полученной.

2.6 Критериальная функция и функция пригодности

В нашем случае критериальной функцией является оптимизируемая функция решаемой задачи.

Функция пригодности позволяет проранжировать хромосомы популяции, присвоив им значение вероятности выживания таким образом, чтобы более подходящая по сравнению с прочими хромосома имела большую вероятность перейти в следующую популяцию. Один из способов построения данной функции заключается в следующем.

Упорядочим все хромосомы текущей популяции по убыванию их «подходящести» (для нашей задачи максимизации — по убыванию значения критериальной функции). Затем выберем случайным образом параметр $a \in (0, 1)$ и запишем формулу пригодности:

$$eval(V_i) = a \times (1 - a)^{i-1}, \quad i = 1, \dots, popsize.$$

Как видно из этой функции, реальные значения критериальной функции здесь не важны, важен только порядок хромосом. Далее эта функция используется в алгоритме отбора хромосом.

2.7 Алгоритм отбора хромосом

Процесс отбора хромосом не зависит от выбора функции пригодности и основывается на так называемом алгоритме рулетки, который наглядно можно представить следующим образом. Пусть у нас есть круглый диск, который может вращаться вокруг собственной оси, и стрелка, указывающая на его центр. Разобьем всю площадь диска на $popsize$ секторов и в каждый сектор положим по одной хромосоме из имеющейся на данный момент популяции. При этом площадь сектора с i -той хромосомой будет составлять

$$\frac{eval(V_i)}{\sum_{j=1}^{popsize} eval(V_j)}$$

долю от общей площади диска.

Раскручиваем рулетку, и когда она останавливается, выбираем хромосому, находящуюся в том секторе, на который указывает стрелка. Делаем копию этой хромосомы, помещаем эту копию в результирующее множество, а исходную хромосому возвращаем на прежнее место на рулетке. Эту операцию мы проделываем $popsize$ раз.

Формально этот алгоритм можно описать так:

- Вычислить кумулятивную «вероятность» отбора q_i для каждой хромосомы V_i :

$$\begin{cases} q_0 = 0, \\ q_i = \sum_{j=1}^i eval(V_j), \quad i = 1, \dots, popsize \end{cases}$$

- Повторить следующие два шага $popsize$ раз и получить $popsize$ копий хромосомом:
 - Сгенерировать число $r \in (0, q_{popsize}]$.
 - Выбрать V_i хромосому, где i определяется по правилу $q_{i-1} < r \leq q_i$.

Следует заметить, что в результате отбора мы получим популяцию, в которой ряд хромосом будут дублироваться. При этом, чем более пригодной была хромосома в исходной популяции, тем больше дубликатов у нее будет в новой популяции.

3. Модельный пример

Исследуем на модельном примере поведение задачи возможностной оптимизации в зависимости от t -нормы, описывающей взаимодействие нечетких параметров.

3.1 Задача уровневой оптимизации с T_W -связанными параметрами

Рассмотрим задачу уровневой оптимизации (1)-(3) с T_W -связанными параметрами при $n = m = 2$ и параметрах $\alpha_0 = 0.5$, $\alpha_1 = 0.3$ и $\alpha_2 = 0.4$:

$$k \rightarrow \max,$$

$$\pi \{a_{01}(\gamma)x_1 + a_{02}(\gamma)x_2 = k\} \geq 0.5,$$

$$\begin{cases} \pi \{a_{11}(\gamma)x_1 + a_{12}(\gamma)x_2 - b_1(\gamma) = 0\} \geq 0.3, \\ \pi \{a_{21}(\gamma)x_1 + a_{22}(\gamma)x_2 - b_2(\gamma) = 0\} \geq 0.4, \\ x \in \mathbb{R}_+^2. \end{cases}$$

Здесь $a_{01}(\gamma)$, $a_{02}(\gamma)$, $a_{11}(\gamma)$, $a_{12}(\gamma)$, $a_{21}(\gamma)$, $a_{22}(\gamma)$, $b_1(\gamma)$ и $b_2(\gamma)$ — триангулярные взаимно T_W -связанные нечеткие величины:

$$\begin{aligned} a_{01}(\gamma) &= Tr(1, 2), & a_{02}(\gamma) &= Tr(5, 3), & a_{11}(\gamma) &= Tr(8, 3), & a_{12}(\gamma) &= Tr(-2, 1), \\ a_{21}(\gamma) &= Tr(1, 2), & a_{22}(\gamma) &= Tr(-5, 4), & b_1(\gamma) &= Tr(14, 7), & b_2(\gamma) &= Tr(0, 5). \end{aligned}$$

Представим эти нечеткие величины в (L, R) -форме. Для триангулярных нечетких величин функции представления форм будут следующими:

$$L(t) = R(t) = \max\{0, 1 - t\}, \quad t \in \mathbb{R}_+^1.$$

Получаем нечеткие величины (L, R) -типа:

$$\begin{aligned} a_{01}(\gamma) &= (1, 1, 1, 1)_{LR}, & a_{02}(\gamma) &= \left(5, 5, \frac{3}{2}, \frac{3}{2}\right)_{LR}, \\ a_{11}(\gamma) &= \left(8, 8, \frac{3}{2}, \frac{3}{2}\right)_{LR}, & a_{12}(\gamma) &= \left(-2, -2, \frac{1}{2}, \frac{1}{2}\right)_{LR}, & b_1(\gamma) &= \left(14, 14, \frac{7}{2}, \frac{7}{2}\right)_{LR}, \\ a_{21}(\gamma) &= (1, 1, 1, 1)_{LR}, & a_{22}(\gamma) &= (-5, -5, 2, 2)_{LR}, & b_2(\gamma) &= \left(0, 0, \frac{5}{2}, \frac{5}{2}\right)_{LR}. \end{aligned}$$

Построим детерминированный эквивалентный аналог задачи. Согласно (8)-(9) имеем:

$$x_1 + 5x_2 + \frac{1}{2} \max \left\{ x_1, \frac{3}{2}x_2 \right\} \rightarrow \max, \quad (10)$$

$$\begin{cases} \max \left\{ x_1, \frac{3}{2}x_2 \right\} \geq 0, \\ 8x_1 - 2x_2 - \max \left\{ \frac{3}{2}x_1, \frac{1}{2}x_2 \right\} \cdot 0.7 \leq 16.45, \\ 8x_1 - 2x_2 + \max \left\{ \frac{3}{2}x_1, \frac{1}{2}x_2 \right\} \cdot 0.7 \geq 11.55, \\ x_1 - 5x_2 - \max \{x_1, 2x_2\} \cdot 0.6 \leq 1.5, \\ x_1 - 5x_2 + \max \{x_1, 2x_2\} \cdot 0.6 \geq -1.5, \\ x \in \mathbb{R}_+^2. \end{cases} \quad (11)$$

Инициализируем генетический алгоритм решения полученной задачи следующими константами:

1. размер популяции $popsizе = 50$,
2. вероятность мутации $P_m = 0.2$,
3. вероятность кроссовера $P_c = 0.8$,
4. константа, участвующая в алгоритме отбора хромосом, $a = 0.05$.

Работа генетического алгоритма представлена на рис.1. Как видно из рис.1, генетический алгоритм уже на 80-й итерации сойдется к оптимуму с точностью до третьего знака после запятой. По оси ординат на приведенном рисунке указаны значения наилучшей хромосомы с точностью до третьего знака после запятой. По оси абсцисс — количество итераций генетического алгоритма.

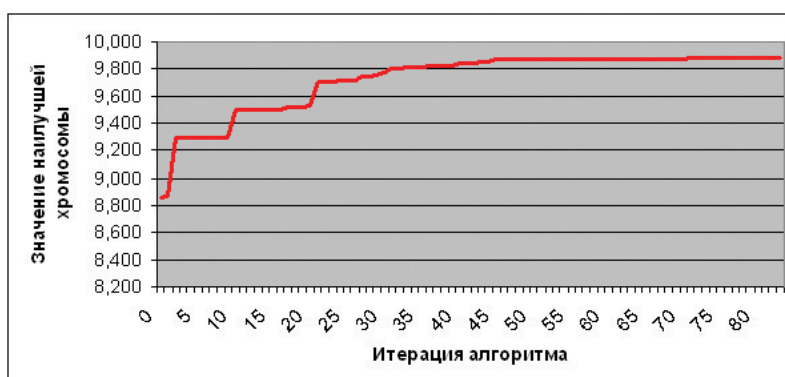


Рис. 1: Работа генетического алгоритма.

Решение задачи, полученное генетическим алгоритмом, следующее:

$$x_1 \approx 2.70206, \quad x_2 \approx 1.16466, \quad F_W(x_1, x_2) \approx 9.87638.$$

На рис. 2 изображена область допустимых решений, задаваемая ограничениями. Полоса ограничений (А) задается вторым и третьим неравенствами системы (11), полоса ограничений (Б) — четвертым и пятым неравенством системы (11). Первое неравенство задает область, покрывающую целиком пересечение первых двух областей, поэтому на рисунке отсутствует.

Пересечение данных областей определяет ту область, в которой работает генетический алгоритм. Крестиком отмечена точка, доставляющая максимум целевой функции.

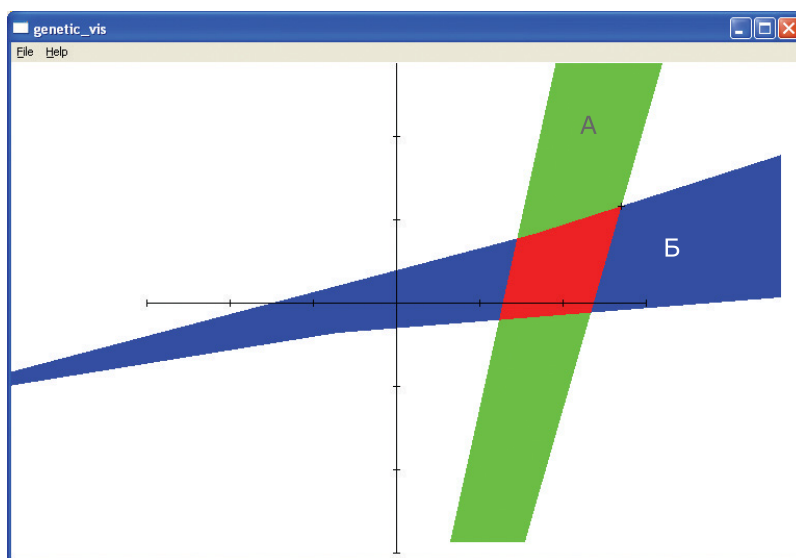


Рис. 2: Область допустимых решений эквивалентного детерминированного аналога задачи (1)-(3) для случая взаимно T_W -связанных параметров.

3.2 Задача уровневой оптимизации с взаимно-минисвязанными параметрами

Рассмотрим задачу в случае, когда все параметры являются взаимно минисвязанными триангулярными нечеткими величинами. Ее детерминированный эквивалентный аналог, согласно [7], будет выглядеть следующим образом:

$$\begin{cases} 1.5x_1 + 5.75x_2 \rightarrow \max, \\ 6.95x_1 - 2.35x_2 \leq 16.45, \\ 9.05x_1 - 1.65x_2 \geq 11.55, \\ 0.4x_1 - 6.2x_2 \leq 1.5, \\ 1.6x_1 - 3.8x_2 \geq -1.5, \\ x \in \mathbb{R}_+^2. \end{cases}$$

Решение полученной задачи линейного программирования:

$$x_1 \approx 2.915, \quad x_2 \approx 1.6223, \quad F_M(x_1, x_2) \approx 13.701.$$

На рис. 2 можно видеть области допустимых решений модельной задачи в случае взаимно-минисвязанных и взаимно T_W -связанных параметров.

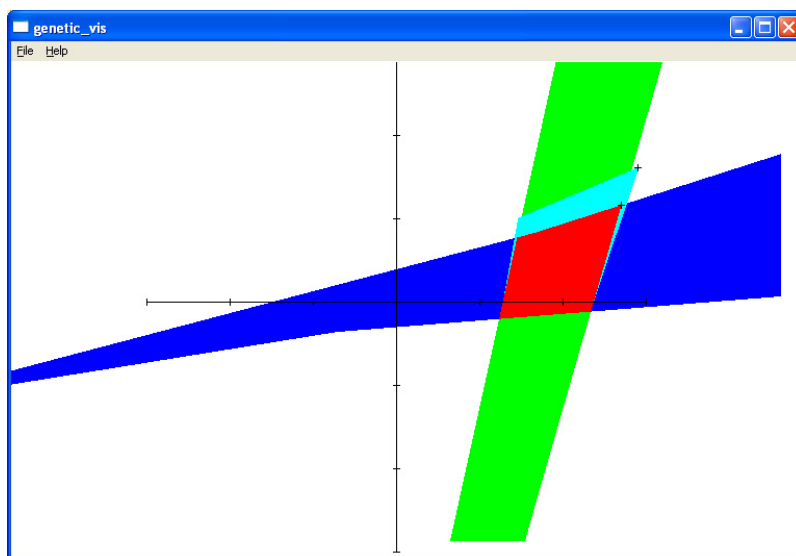


Рис. 3: Области допустимых решений эквивалентных детерминированных аналогов задачи (1)-(3) для случая взаимно T_W -связанных и взаимно минисвязанных параметров.

Как видно, в случае взаимно T_W -связанных параметров область получается более компактной, а решение, соответственно, менее «размытым», чем в случае взаимной минисвязанности. Это можно доказать строго математически.

Действительно, обозначим через $X_M^{\alpha_i}$ множество решений x , удовлетворяющих i -му ограничению модели (3), при условии, что все $a_{ij}(\gamma)$ и $b_i(\gamma)$ — минисвязанные:

$$X_M^{\alpha_i} = \left\{ x : \pi \{ f_i(x, \gamma) = 0 \} \geq \alpha_i \right\}, \quad a_{ij}(\gamma) \text{ и } b_i(\gamma) \text{ — минисвязанные,}$$

а через $X_W^{\alpha_i}$ — при условии, что все $a_{ij}(\gamma)$ и $b_i(\gamma)$ — T_W -связанные:

$$X_W^{\alpha_i} = \left\{ x : \pi \{ f_i(x, \gamma) = 0 \} \geq \alpha_i \right\}, \quad a_{ij}(\gamma) \text{ и } b_i(\gamma) \text{ — } T_W\text{-связанные.}$$

Может быть доказана следующая теорема.

Теорема 1.

$$\bigcap_{i=1}^m X_W^{\alpha_i} \subseteq \bigcap_{i=1}^m X_M^{\alpha_i}.$$

Подобную теорему можно доказать и для модели критерия. Обозначим через $F_M^{\alpha_0}(x)$ множество значений критерия, удовлетворяющих модели (2), при условии, что все $a_{0j}(\gamma)$ — взаимно минисвязанные, а через $F_W^{\alpha_0}(x)$ — множество значений критерия, удовлетворяющих модели (2), при условии, что все $a_{0j}(\gamma)$ — взаимно T_W -связанные.

Можно доказать также следующую теорему.

Теорема 2. $F_W^{\alpha_0}(x) \subseteq F_M^{\alpha_0}(x)$.

4. Обсуждение

Для проведения сравнительного анализа, рассмотрим следующую задачу линейного программирования, в которой все параметры являются четкими:

$$\begin{cases} x_1 + 5x_2 \rightarrow \max, \\ 8x_1 - 2x_2 - 14 = 0, \\ x_1 - 5x_2 = 0, \\ x \in \mathbb{R}_+^2. \end{cases}$$

Решение данной задачи линейного программирования следующее:

$$x_1 \approx 1.842, \quad x_2 \approx 0.368, \quad F_C(x_1, x_2) \approx 3.682.$$

Сведем результаты решения трех задач в таблицу:

параметры	x_1	x_2	$F_T(x_1, x_2)$
четкие	1.842	0.368	3.682
взаимно T_W -связанные	2.70206	1.16466	9.87638
взаимно минисвязанные	2.915	1.6223	13.701

Таблица 1

Результаты решения модельных примеров, представленные в таблице 1, подтверждают теоретические результаты, содержащиеся в предыдущем разделе.

Можно выдвинуть гипотезу, что для любой t -нормы T ($T_W \leq T \leq T_M$) значение целевого функционала $F_T(x)$ удовлетворяет следующему неравенству:

$$F_W(x) \leq F_T(x) \leq F_M(x).$$

Однако данное предположение требует дальнейших исследований.

Список литературы

- [1] Д.Дюбуа, А.Прад, Теория возможностей / Пер. с франц. М.: Радио и связь, 1990.
- [2] И.С.Солдатенко, О методе решения одной задачи возможностного программирования в случае T_W -связанных параметров, // Сборник научных трудов IV-й Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (28-30 мая 2007г.) том 1, с.231-238.

-
- [3] B.Liu, Uncertain programming, Wiley and Sons, New York, 1999.
 - [4] S.Nahmias, Fuzzy Variables, Fuzzy Sets and Systems 1 (1978) 97-110.
 - [5] Wang Xizhao, Ha Minghu, Note on maxmin μ/E estimation, Fuzzy Sets and Systems 94 (1998) 71-75.
 - [6] A.V.Yazenin Fuzzy and Stochastic Programming, Fuzzy Sets and Systems 22 (1987) 171-180.
 - [7] A.V.Yazenin On the problem of possibilistic optimization, Fuzzy Sets and Systems 81 (1996) 133-140.
 - [8] A.V.Yazenin, M.Wagenknecht, Possibilistic optimization, Brandenburgische Technische Universitat, Cottbus, Germany, 1996.