

## Подход к уменьшению времени работы модифицированной модели Голдберга при решении неоднородной минимаксной задачи

*В.Г. Кобак<sup>1</sup>, А.Г. Жуковский<sup>2</sup>, А.П. Кузин<sup>1</sup>, А.Н. Тхазаплизева<sup>1</sup>*

<sup>1</sup>*Донской государственной технической университет, Ростов-на-Дону*

<sup>2</sup>*Северо-Кавказский филиал Московского технического университета связи и информатики*

**Аннотация:** В статье рассматривается проблема решения неоднородной минимаксной задачи, характерной для теории расписаний. Данная задача является NP-полной и для нее не существует точного алгоритма решения, имеющего полиномиальное время для задач большой размерности. В качестве метода решения данной задачи рассматривается модифицированная модель Голдберга. Модель Голдберга рассматривается с несколькими кроссоверами и наиболее эффективной мутацией. При определенных параметрах (большое количество особей и повторов) модифицированная модель Голдберга получает решение за достаточно долгое время, поэтому в статье подробно анализируется один из подходов по уменьшению времени работы без потери точности. Так как аналитически произвести расчеты крайне затруднительно и практически невозможно в работе был поставлен вычислительный эксперимент. В результате вычислительного эксперимента, в таблицах приводится сравнение эффективности работы модифицированной модели Голдберга после применения НТ технологии. Применение НТ технологии приводит к существенному уменьшению временных затрат.

**Ключевые слова:** одноточечный кроссовер, двухточечный кроссовер генетический алгоритм, модифицированная модель Голдберга, мутация, минимаксная задача, теория расписаний, особь, поколение, hyper-threading.

### Введение

Одними из наиболее часто решаемых задач теории расписаний являются NP-полные задачи, для которых практически невозможно подобрать решение за полиномиально быстрое время. К таким задачам относится также и неоднородная минимаксная задача. Разработка различных методов, позволяющих получить близкое к оптимальному приближенное решение, является актуальной проблемой. Такие решения находятся в том числе и с использованием генетических алгоритмов т с их различными моделями.

Основным механизмом эволюции является сочетание генетического механизма передачи наследственности при помощи, механизма мутаций. Такой механизм обеспечивает разнообразие видов, и естественного отбора,

который позволяет с течением времени сформировать наиболее приспособленные для данной среды популяции особей. Более приспособленные особи имеют большую вероятность передать свою наследственную информацию. Наследственная информация в виде хромосомы полностью определяет развитие особи в ее жизненном цикле, и с помощью обмена генами хромосом происходит передача наследственной информации потомкам. Случайные изменения в генофонд вносятся во время мутаций, если новые признаки увеличивают приспособленность особи, то такие признаки, возможно, закрепятся и перейдут к потомкам.

### Постановка задачи

В терминах теории расписаний неоднородная минимаксная задача может быть сформулирована следующим образом. Имеется система обслуживания, состоящая из  $N$  независимых устройств  $P = \{p_1, p_2, \dots, p_n\}$ . На обслуживание поступает конечный поток  $M$  – множество независимых параллельных заданий (функциональных операторов)  $T = \{t_1, t_2, \dots, t_m\}$ .  $\tau(t_i p_j)$  – длительность обслуживания задания  $t_i$  устройством  $p_j$ , определяется матрицей  $T_\tau$ . Приборы в общем случае не идентичны, задание  $t_i$  может быть обслужено любым из устройств, и устройство  $p_j$  может обрабатывать одновременно не более одного задания. Необходимо определить такое распределение заданий по устройствам без прерываний, чтобы время выполнения всей совокупности заданий было минимальным. Критерием минимизации времени выполнения заданий, является минимаксный критерий, который определяется в следующем виде:  $f = \max_{1 \leq j \leq n} f_j \rightarrow \min$ , где

$$f_j = \sum_{\tau(t_i p_j) \in T} \tau(t_i p_j) - \text{время завершения работы процессора } p_j [1,2].$$

В данной работе для решения неоднородной минимаксной задачи будет рассматриваться модифицированная модель Голдберга, являющаяся одним из видов моделей генетических алгоритмов (далее ГА). Модифицированная модель Голдберга отличается от классической модели Голдберга, тем, что используются различные кроссоверы и мутации и бинарный турнирный отбор в следующее поколение. Более детально с модифицированной моделью можно ознакомиться в работе [3-5].

В данной работе для исследования рассматривается как классический одноточечный кроссовер, изображенный на рисунке 1, так и двухточечный кроссовер изображенный на рисунке 2.

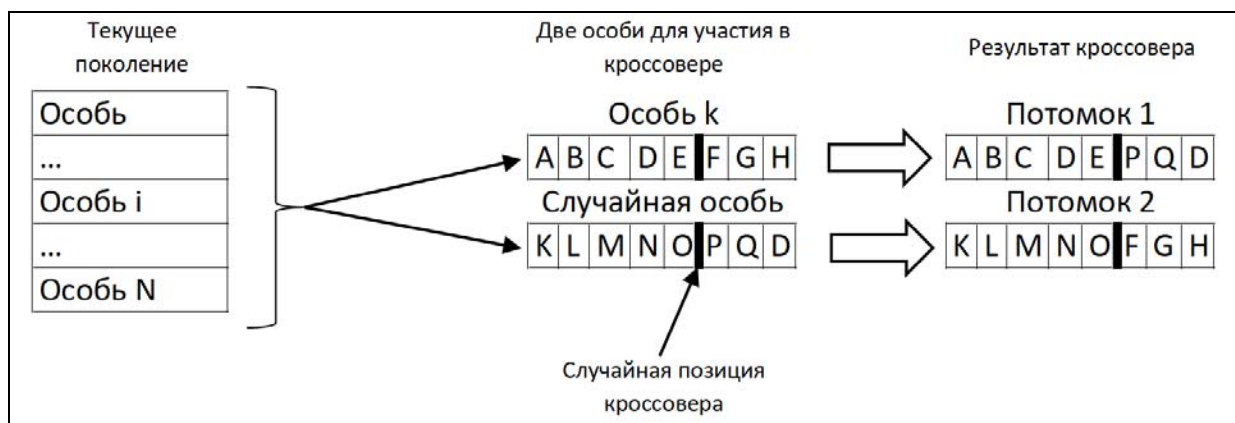


Рис. 1 – Классический одноточечный кроссовер.

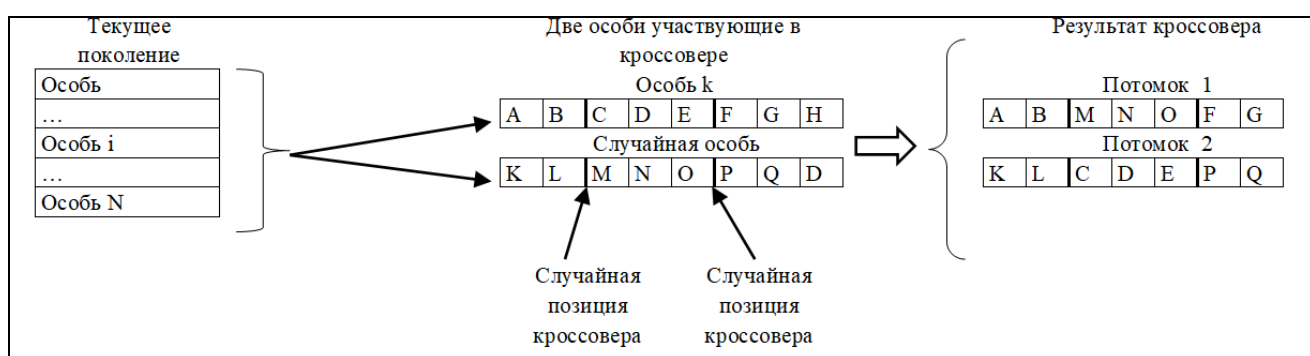


Рис. 2 – Двухточечный кроссовер.

В работе [2,3,4] были исследованы различные модификации мутаций, из всего спектра которых была выбрана одна как наиболее эффективная, а именно простая мутация, схематически изображенная на рисунке 3:

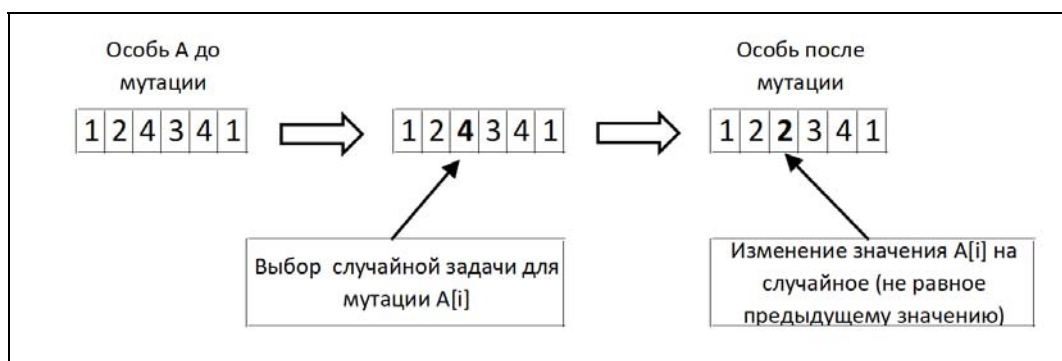


Рис. 3 – Простая мутация.

### Вычислительный эксперимент

С ростом размерности минимаксной неоднородной задачи увеличивается время необходимое для получения решения при использовании генетического алгоритма. При проведении вычислительного эксперимента была заполнена таблица 1, содержащая оценку роста временных затрат в зависимости от размерности задачи, а также, как показано в [6-8] от количества особей и повторов.

Таблица 1

Оценка временных затрат для получения решения.

	Особь и повторы	Одноточечный кроссовер	Двухточечный кроссовер
<b>4*501</b>			
<b>Время (с.)</b>	<b>400*400</b>	215	231
	<b>800*800</b>	626	666
	<b>1600*1600</b>	1600	1956
<b>5*501</b>			
<b>Время (с.)</b>	<b>400*400</b>	335	337
	<b>800*800</b>	800	816
	<b>1600*1600</b>	2336	2664
<b>6*501</b>			
<b>Время (с.)</b>	<b>400*400</b>	507	547
	<b>800*800</b>	1294	1140
	<b>1600*1600</b>	3339	3752

Для вычислительных экспериментов был использован 4 ядерный процессор с поддержкой технологии Hyper-Threading, за счет которой реализуется поддержка работы с 8 потоками.

Hyper-Threading – это технология, которая была разработана компанией Intel. Суть технологии заключается в том, что она позволяет одному ядру процессора выполнять два потока данных одновременно. В ходе многочисленных исследований было выяснено, что обычный процессор зачастую при решении задач использует не более 70% всей своей вычислительной мощности. Было принято решение использовать технологию, позволяющую при простом вычислительном блоке, нагружать его работой с другим потоком. Данный подход позволяет увеличить производительность одного ядра от 10 до 80% в зависимости от особенностей задачи. Производительность таких виртуальных потоков гораздо ниже полноценного ядра. Использование виртуальных потоков позволяет добиться большей эффективности вычислительной мощности, путем загрузки процессора на 100% вычислениями, сокращая время простоя, при этом производительность в некоторых задачах может повыситься на 50% [9,10].

В ходе проведения вычислительного эксперимента был проведен ряд расчетов. В рамках исследования оценивались такие параметры как время поиска решения, среднее и минимальное значения, полученные в ходе эксперимента. Все эксперименты проводились для 100% вероятности кроссовера и мутации.

В рамках вычислительного эксперимента решение неоднородной минимаксной задачи генетическим алгоритмом была модифицирована для многопоточных вычислений. Для оценки времени получаемого решения

были взяты 3 варианта размерности задачи: 4 устройства и 501 задача; 5 устройств и 501 задача; 6 устройств и 501 задача.

Таблица №2

Сравнение эффективности применения многопоточных вычислений для однопоточного и двухточечного кроссовера.

Особь и повторы	Параметры	Однопоточный кроссовер				Двухточечный кроссовер			
		1 поток	2 потока	4 потока	8 потоков	1 поток	2 потока	4 потока	8 потоков
<b>4*501</b>									
400*400	мин	3459	3450	3430	3442	3369	3379	3393	3389
	сред	3486,1	3498,6	3495	3479,8	3408	3402,5	3406,5	3408,8
	время	215	114	112	84	231	147	104	89
800*800	мин	3411	3424	3417	3414	3365	3363	3373	3372
	сред	3438,8	3447,4	3450,3	3445,1	3380,3	3375,6	3382,3	3385,9
	время	626	339	263	193	666	448	279	211
1600*1600	мин	3399	3389	3398	3395	3359	3361	3359	3362
	макс	3414,3	3405,1	3418,5	3415	3368,7	3370,1	3369,5	3376,6
	время	1600	931	619	471	1956	1039	743	580
<b>5*501</b>									
400*400	мин	2742	2749	2737	2746	2670	2693	2674	2693
	сред	2784,6	2774,9	2757,5	2810,3	2712,3	2716,7	2707,3	2719,4
	время	335	188	172	106	337	187	156	1320
800*800	мин	2725	2718	2716	2705	2684	2667	2676	2677
	сред	2755,8	2740,8	2752,2	2772,5	2692,5	2682,3	2688,6	2694,1
	время	800	472	358	296	816	625	386	330
1600*1600	мин	2693	2695	2692	2694	2662	2662	2662	2666
	сред	2719	2719,5	2712,5	2715,9	2672,7	2672,9	2669,5	2679,1
	время	2336	1265	900	747	2664	1473	1045	780
<b>6*501</b>									
400*400	мин	2274	2273	2282	2288	2210	2209	2213	2217
	сред	2303,9	2301,2	2304,7	2327,2	2228,4	2238,4	2244,5	2243,7
	время	507	291	233	149	547	287	189	158
800*800	мин	2259	2253	2254	2250	2207	2205	2205	2208
	сред	2276,6	2287,7	2284	2293,6	2226,8	2226,8	2226	2225,5
	время	1294	644	529	346	1140	668	484	392
1600*1600	мин	2234	2243	2235	2242	2192	2201	2196	2192
	сред	2247,7	2263,3	2263,8	2255,8	2207,1	2211,4	2208	2211,2

	время	3339	1702	1123	883	3752	1808	1173	1024
--	-------	------	------	------	-----	------	------	------	------

Из таблицы 2 видно, что наибольший выигрыш по времени дает использование 8 потоков. Для оценки эффективности дальнейшего увеличения потоков был проведен эксперимент по сравнению времени получения решения при использовании 8 и 16 потоков, результаты которого представлены в таблице 3.

Таблица №3

## Сравнение эффективности 8 и 16 потоков.

Особи и повторы	Одноточечный кроссовер		Двухточечный кроссовер	
	8 потоков	16 потоков	8 потоков	16 потоков
<b>4*501</b>				
<b>400*400</b>	84	80	89	85
<b>800*800</b>	193	196	211	214
<b>1600*1600</b>	471	504	580	632
<b>5*501</b>				
<b>400*400</b>	106	123	132	116
<b>800*800</b>	296	286	330	295
<b>1600*1600</b>	747	724	780	757
<b>6*501</b>				
<b>400*400</b>	149	176	158	151
<b>800*800</b>	346	377	392	368
<b>1600*1600</b>	883	921	1024	1038

**Выводы**

1. С увеличением количества особей и количества повторов при использовании различных кроссоверов модифицированная модель Голдберга дает как лучшие средние результаты, так и лучшее отдельное решение, наиболее приближенное к точному.

2. С увеличением количества особей и количества повторов при использовании различных кроссоверов резко увеличивается время получения решения модифицированной моделью Голдберга.

3. Использование технологии Hyper-Threading при решении неоднородной минимаксной задачи модифицированной моделью Голдберга позволяет в разы уменьшить время получения решения, что позволяет решать задачи большей размерности.

4. Использование технологии Hyper-Threading при решении неоднородной минимаксной задачи модифицированной моделью Голдберга позволяет определиться с количеством потоков равным 8. Дальнейшее увеличение потоков не приводит к реальному уменьшению расчетного времени для обоих кроссоверов.

### Литература

1. Кобак В.Г., Жуковский А.Г., Кузин А.П. Исследование применения односточечного кроссовера при решении неоднородной минимаксной задачи //Инженерный вестник Дона, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4714/.

2. Кобак В.Г., Жуковский А.Г., Кузин А.П. Исследование модификаций турнирного отбора при решении неоднородной минимаксной задачи модифицированной моделью Голдберга. //Инженерный вестник Дона, 2018, №2. URL: ivdon.ru/ru/magazine/archive/N2y2018/4962.

3. Титов Д.В., Кобак В.Г. Анализ подходов к улучшению результатов работы генетического алгоритма при решении однородной минимаксной задачи. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей Всерос. научно-техн. конф.– Пенза: ПДЗ, 2008, С. 76-78.

4. Кобак В.Г., Поркшеян В.М. и Кузин А.П. Использование различных вариантов мутации при решении неоднородной минимаксной





задачи модифицированной моделью Голдберга // Научно практический журнал «Аспирант», 2017, №10, С. 26-29.

5. Каширина И.Л. Введение в эволюционное моделирование. Воронеж, 2007, 40 С.

6. Кобак В. Г. Методический подход к улучшению работы генетического алгоритма в однородной минимаксной задаче. Вестник Дон. гос. техн. ун-та, 2010, Т 10, № 4(47), С. 474–479.

7. Аль-Хулайди А.А., Чернышев Ю.О. Разработка параллельного алгоритма нахождения оптимального решения транспортной задачи на кластере // Инженерный вестник Дона, 2011, №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445/.

8. Нетёсов А.С. Эволюционно-генетический подход к решению задач оптимизации. Сравнительный анализ генетических алгоритмов с традиционными методами оптимизации // Инженерный вестник Дона, 2011, №3. URL: ivdon.ru/ru/magazine/archive/n3y2011/459/.

9 Desktop Products Group. Hyper-threading technology architecture and microarchitecture // Intel Technology Journal, 2002, Q1. URL: cs.virginia.edu/~mc2zk/cs451/vol6iss1\_art01.pdf.

10 Wessam M. Hassanein. Analyzing the Effects of Hyperthreading on the Performance of Data Management Systems // International Journal of Parallel Programming, 2008, V. 36, PP 206–225.

### References

1. Kobak V.G., Zhukovskiy A.G., Kuzin A.P. Inženernyj vestnik Dona (Rus), 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4714/.

2. Kobak V.G., Zhukovskiy A.G., Kuzin A.P. Inženernyj vestnik Dona (Rus), 2018, №2. URL: ivdon.ru/ru/magazine/archive/N2y2018/4962.

3. Titov D.V., Kobak V.G. Problemy informatiki v obrazovanii, upravlenii, ekonomike i tekhnike, Penza, 2008, PP. 76-78.



4. Kobak V.G., Porkshejan V.M., Kuzin A.P. Nauchno prakticheskij zhurnal «Aspirant», 2017, №10, PP. 26-29.
5. Kashirina I.L. Vvedeniye v evolyutsionnoye modelirovaniye [Introduction to evolutionary modeling]. Voronezh, 2007, 40 P.
6. Kobak V.G. Vestnik Donskogo Gosudarstvennogo Tekhnicheskogo Universiteta, 2010, T 10, № 4(47), PP. 474–479.
7. Al'-Khulaydi A.A., Chernyshev YU.O. Inženernyj vestnik Dona (Rus), 2011, №2. URL: [ivdon.ru/ru/magazine/archive/n2y2011/445/](http://ivdon.ru/ru/magazine/archive/n2y2011/445/).
8. Netosov A.S. Inženernyj vestnik Dona (Rus), 2011, №3. URL: [ivdon.ru/ru/magazine/archive/n3y2011/459/](http://ivdon.ru/ru/magazine/archive/n3y2011/459/).
9. Desktop Products Group. Intel Technology Journal, 2002, Q1. URL: [cs.virginia.edu/~mc2zk/cs451/vol6iss1\\_art01.pdf](http://cs.virginia.edu/~mc2zk/cs451/vol6iss1_art01.pdf).
10. Wessam M. Hassanein. International Journal of Parallel Programming, 2008, V. 36, PP. 206–225.