Инженерный вестник Дона, №1 (2018)
ivdon.ru/ru/magazine/archive/n1y2018/4714

Исследование применения одноточечного кроссовера при решении неоднородной минимаксной задачи

В.Г. Кобак[1], А.Г. Жуковский[2], А.П. Кузин[1]

[1] *Донской государственный технический университет*
[2] *Северо-Кавказский филиал Московского технического университета связи и информатики*

Аннотация: В статье рассматривается проблема решения минимаксной задачи, характерной для теории расписаний. В качестве возможного метода решения данной задачи рассматривается модифицированная модель Голдберга, являющаяся одной из разновидностей генетических алгоритмов. Описывается сравнение эффективности работы данной модели на основе оценки точности полученных результатов при использовании стандартного кроссовера, для различных видов мутаций и параметров генетического алгоритма.
Ключевые слова: одноточечный кроссовер, генетический алгоритм, модифицированная модель Голдберга, мутация, минимаксная задача, теория расписаний, элитная особь, особь, поколение.

При решении ряда определенных задач существенное влияние создает порядок, в котором функциональные операторы выполняются. Решением такого рода экстремальных задач комбинаторного типа занимается теория расписаний. В рамках теории расписаний исследуются методы, позволяющие упорядочить или другими словами определить последовательность выполнения совокупности работ таким образом, чтобы время выполнения задачи в целом было минимальным. Задача получения оптимального упорядочивания работ относится к NP-полным задачам, трудоемкость решения которой определяется как $O(n^m)$, где O - временная асимптотическая сложность алгоритма, а n, m - целое число больше единицы. Практическая актуальность решения таких задач определяется возможностью экономии машинного времени.

В терминах теории расписаний распределительная задача может быть сформулирована следующим образом. Имеется система обслуживания, состоящая из N независимых устройств $P = \{p_1, p_2, ..., p_n\}$. На обслуживание поступает конечный поток $M-$ множество независимых параллельных заданий (функциональных операторов) $T=\{t_1,t_2,...,t_m\}$. $\tau(t_i p_j)-$ длительность обслуживания задания t_i устройством p_j, определяется матрицей T_τ. 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. Критерий минимизации времени завершения обслуживания заданий, является минимаксным критерием и определяется в следующем виде: $f=\max\limits_{l\le j\le n} f_j \to \min$, где $f_j=\sum\limits_{\tau(t_i p_j)\in T}\tau(t_i p_j)$ – время завершения работы процессора p_j [1, 4].

Для решения поставленной задачи используются различные алгоритмы, позволяющие получить точное или приближенное решение. В данной работе в качестве базового алгоритма для решения неоднородной минимаксной задачи возьмем модифицированную модель Голдберга p_j [2, 3], являющеюся одним из видов моделей генетических алгоритмов (далее ГА). Модифицированная модель Голдберга отличается от классической модели Холланда [5, 6, 7], тем, что использует турнирный отбор особей в новое поколение, который позволяет улучшить результаты работы алгоритма с различными модификациями мутации при одноточечном кроссовере.

Модифицированную модель Голдберга можно описать в виде последовательности следующих шагов:

Шаг 1. Формируется начальное поколение, состоящее из заданного числа особей.

Шаг 2. Турнирный отбор особей и применение ГА операторов кроссовера и мутации с заданной вероятностью для создания нового поколения.

Шаг 3. Проверка условия конца работы алгоритма, которая обычно заключается в неизменности лучшего решения в течение заданного числа поколений. Если проверка прошла неуспешно, то переход на шаг 2.

Шаг 4. Лучшая особь выбирается как найденное решение [8,9,10].

Графически функционирование модифицированной модели Годберга можно изобразить на рис.1. Лучшая особь выбирается и ставится в следующее поколение. Процесс повторяется до тех пор, пока лучшая особь в поколении не повторится заданное разработчиком количество раз.

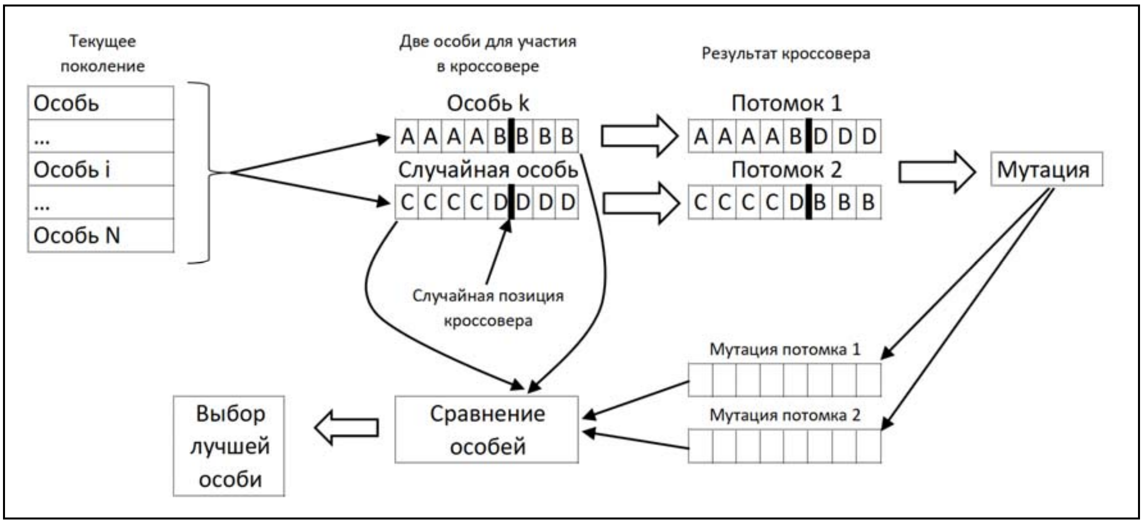

Рис. 1 – Схема функционирования модифицированной модели Голдберга.

В данной работе для исследования рассматривается классический одноточечный кроссовер, изображенный на рис.2.

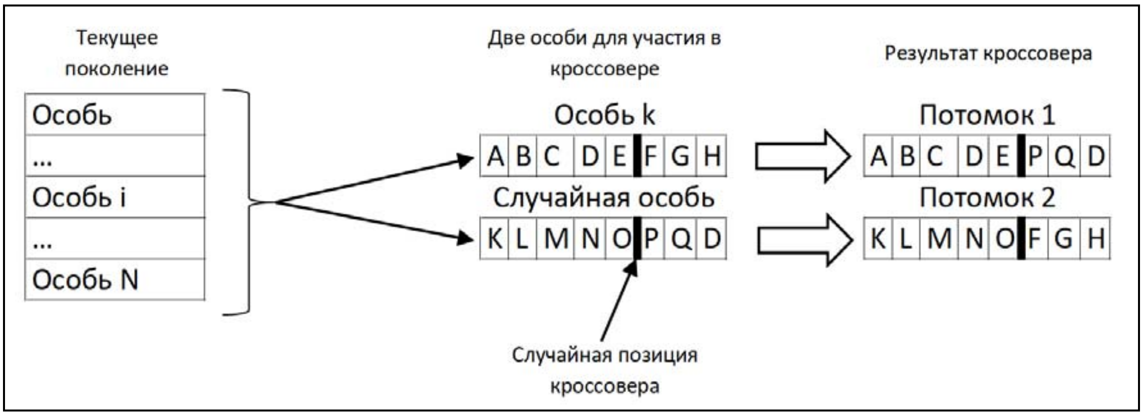

Рис. 2 – Классический одноточечный кроссовер.

В работе [3, 9, 10] были исследованы различные модификации мутаций, из всего спектра которых были выбраны для исследований следующие:

1. Простая мутация. Для выбранной особи *A* случайно выбирается номер задачи *i*. Генерируется случайное число *z* в диапазоне от 1 до количества возможных устройств. Проверяется, что полученное число *z* не совпадает со значением *A[i]*, если совпадает, то *z* генерируется заново, иначе *A[i]* присваивается значение *z*. Пример простой мутации изображен на рис.3.

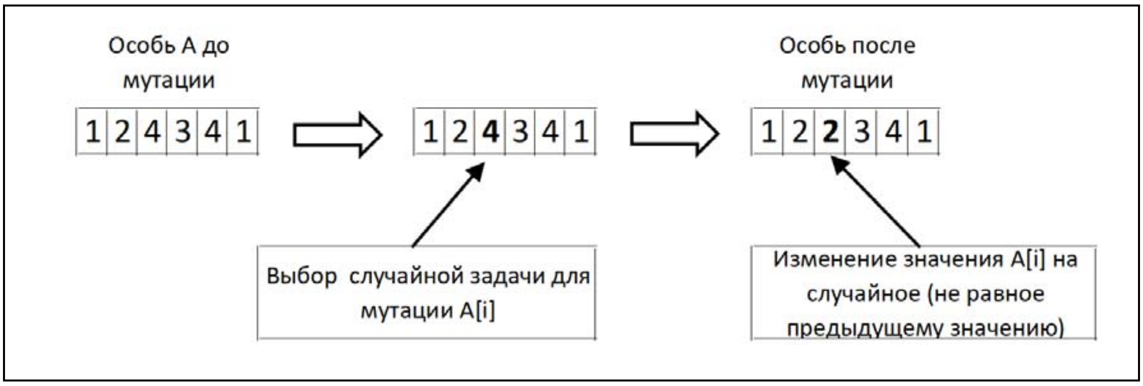

Рис. 3 – Пример простой мутации.

2. Простая мутация с возможным повтором. Аналогична простой мутации, но отличается тем что в данной мутации отсутствует проверка совпадения значения *z* со значением *A[i]*.

3. Раздельная двухбитная мутация. Выбранной особи *A* случайно выбирается номер задачи *i*. Значение *A[i]* рассматривается как массив из 8 бит. Случайно выбираются два бита значения которых инвертируется. В результате получается новое значение *A[i]*. Проверка того, на каком устройстве должна исполняться задача *A[i]*, выполняется следующим образом: диапазон значений 0.255 разбивается на равные интервалы, количество которых равно количеству устройств; порядковый номер интервала соответствует номеру устройства; проверяется какому интервалу

принадлежит значение *A[i]*. Пример двухбитной мутации изображен на рис.4.

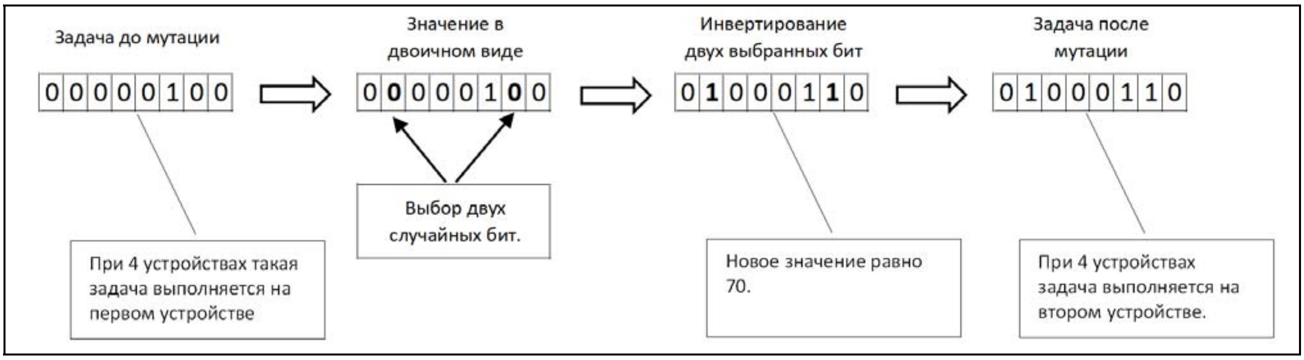

Рис. 4 – Пример раздельной двухбитной мутации.

В данной работе рассмотрим зависимость того как влияет вероятность кроссовера (важнейший параметр генетического алгоритма) на точность решения неоднородной минимаксной задачи.

В связи с тем, что аналитически решить эту задачу крайне проблематично, если вообще возможно, в рамках исследования алгоритмов были поставлены вычислительные эксперименты, позволяющие собрать статистику решений алгоритмами.

Для проведения вычислительного эксперимента было написано программное средство на современном языке программирования C# в среде разработки Microsoft Visual Studio 2017.

Данное программное средство позволяет автоматизировать процесс получения сравнительных характеристик результатов работы генетического алгоритма при различных параметрах.

При проведении вычислительного эксперимента был использован персональный компьютер под управлением Windows 10 Pro x64. В качестве аппаратного обеспечения использовался компьютер со следующей конфигурацией: четырех ядерный процессор Intel Core i7-7700k, 16 гигабайт оперативной памяти формата DDR4. Данная аппаратная система была выбрана в связи с тем, что процессор поддерживает одновременно 8 потоков обработки данных и в связи с этим появляется возможность проводить параллельные вычисления для задач больших размерностей.

Во время проведения вычислительного эксперимента рассматривалась задача со следующими параметрами: 3 устройства, 101 задача, вероятность мутации 100%, количество экспериментов 50. Эксперименты проводились в двух вариантах, при использовании одной элитной особи и без нее. Решение данной задачи стандартными методами является чрезвычайно времязатратным.

Оценивались такие параметры как среднее и минимальное значения, полученные в ходе эксперимента.

Результаты эксперимента при 500 особях и 500 повторах приведены в таблице №1.

Таблица №1

Сравнение полученных результатов при 500 особях и 500 повторов

Количество повторов - 500 Количество особей - 500	Вероятность кроссовера, %	Простая мутация		Простая мутация с возможным повтором		Раздельная двухбитная мутация	
		Мин.	Ср.	Мин.	Ср.	Мин.	Ср.
Без элиты	0	982	991,54	983	990,94	979	991,18
С элитой	0	980	991,46	975	992,28	983	991,96
Без элиты	25	929	936,2	931	941,4	930	939,9
С элитой	25	928	935,78	933	941,48	929	940,56
Без элиты	50	927	935,9	932	940,88	925	939,46
С элитой	50	927	935,84	931	940,42	930	941,24
Без элиты	75	929	937,66	931	940,36	929	940,16
С элитой	75	928	935,52	930	940,3	929	939,88
Без элиты	100	925	935,56	932	939,54	929	941,16
С элитой	100	930	935,56	931	940,58	933	940,82

Результаты эксперимента при 1000 особях и 1000 повторах приведены в таблице №2.

Таблица №2

Сравнение полученных результатов при 1000 особях и 1000 повторов

Количество повторов 1000 Количество особей 1000	Вероятность кроссовера, %	Простая мутация		Простая мутация с возможным повтором		Раздельная двухбитная мутация	
		Мин.	Ср.	Мин.	Ср.	Мин.	Ср.
Без элиты	0	983	988,96	981	989,52	980	988,9
С элитой	0	984	989,92	978	989,76	982	989,46
Без элиты	25	925	929,88	928	934,84	925	931,74
С элитой	25	924	929,7	928	934,74	926	932,02
Без элиты	50	926	930,3	926	933,02	927	933,04
С элитой	50	923	929,52	927	932,92	926	932,12
Без элиты	75	924	929,08	926	933,82	926	932,28
С элитой	75	925	930,16	928	932,6	926	931,72
Без элиты	100	924	929,38	928	932,96	926	932,04
С элитой	100	924	929,16	927	933,22	925	931,74

Результаты эксперимента при 1500 особях и 1500 повторах приведены в таблице №3.

Таблица №3

Сравнение полученных результатов при 1500 особях и 1500 повторов

Количество повторов - 1500 Количество особей 1500	Вероятность кроссовера, %	Простая мутация		Простая мутация с возможным повтором		Раздельная двухбитная мутация	
		Мин.	Ср.	Мин.	Ср.	Мин.	Ср.
Без элиты	0	978	988,1	981	989,24	983	988,12
С элитой	0	981	988,7	981	989,12	982	988,2
Без элиты	25	922	926,74	927	931,08	924	929,2
С элитой	25	923	926,34	927	931,62	923	929,12
Без элиты	50	923	926,64	925	930,18	923	928,96
С элитой	50	923	927,1	924	929,88	924	928,94
Без элиты	75	923	926,38	924	929,44	924	929,18
С элитой	75	923	926,64	926	930,16	924	929,12
Без элиты	100	923	926,86	925	929,8	923	928,58
С элитой	100	923	927,06	925	929,32	924	928,26

Таким образом, проанализировав результаты, приведенные в таблицах 1-3, можно сделать несколько выводов:

1) Чем выше вероятность кроссовера, тем более качественными поучаются как средние результаты, так и лучшие решения.

2) Базовые параметры генетического алгоритма (количество особей и количество повторов наилучшего решения) влияют на качество решения при использовании модифицированной модели Голдберга. Чем большие значения они принимают, тем более ближе к оптимуму получаются как средние результаты, так и лучшие решения.

3) При малых значениях кроссовера использование элитной особи дает положительный эффект, но при больших значениях кроссовера результаты практически одинаковы.

Литература

1. Головкин Б.А. Расчет характеристик и планирование параллельных вычислительных процессов. Москва: Радио и связь, 1983. С. 216.

2. Кобак В.Г., Титов Д.В. Исследование турнирного отбора в генетическом алгоритме для решения однородной минимаксной задачи // Математические методы в технике и технологиях — ММТТ — 21: сб. трудов Междунар. науч. конф. — Саратов. 2008. №.2. С. 12.

3. Кобак В.Г., Поркшеян В.М., Кузин А.П. Использование различных вариантов мутации при решении неоднородной минимаксной задачи модифицированной моделью Голдберга // Научно практический журнал «Аспирант». 2017. №10. С. 26-29.

4. 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/.

5. 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/.

6. Курейчик В. М., Кныш Д. С. Параллельный генетический алгоритм. Модели и проблемы построения // Интегрированные модели и мягкие вычисления в искусственном интеллекте: сб. науч. тр. V Междунар. науч.-практ. конф., Москва: Физматлит, 2009. С. 41-51.

7. Goldberg D. Genetic Algorithms In Search, Optimization, and Machine Learning. USA: Addison-Wesley Publishing Company, Inc., 1989. pp. 28-33.

8. Affenzeller M., Wagner S., Winkler S., Beham A. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. USA: CRC Press, 2009. P. 364.

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

10. Панченко Т. В. Генетические алгоритмы. Астрахань: Астраханский университет, 2007. С. 87.

References

1. Golovkin B.A. Raschet kharakteristik i planirovaniye parallel'nykh vychislitel'nykh protsessov [Characteristics calculation and parallel computing processes planning]. Moscow: Radio i svyaz', 1983. P. 216.

2. Kobak, V. G., Titov D.V. Matematicheskie metody v tehnike i tehnologijah MMTT 21. Saratov, 2008. №.5. P. 12.

3. Kobak V.G., Porkshejan V.M., Kuzin A.P. Nauchno prakticheskij zhurnal «Aspirant». 2017. №10. pp. 26-29.

4. Al'-Khulaydi A.A., Chernyshev YU.O. Inženernyj vestnik Dona (Rus), 2011, №2. URL: ivdon.ru/ru/magazine/archive/n2y2011/445/.

5. Netosov A.S. Inženernyj vestnik Dona (Rus), 2011, №3. URL: ivdon.ru/ru/magazine/archive/n3y2011/459/.

6. Kureychik V. M., Knysh D. S. Integrirovannyye modeli i myagkiye vychisleniya v iskusstvennom intellekte. Moscow: Fizmatlit, 2009. pp. 41-51.

7. Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. USA: Addison-Wesley Publishing Company, Inc., 1989. pp. 28-33.

8. Affenzeller M., Wagner S., Winkler S., Beham A. Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications. USA: CRC Press, 2009. P. 364.

9. Kashirina I.L. Vvedeniye v evolyutsionnoye modelirovaniye [Introduction to evolutionary modeling]. Voronezh, 2007. P. 40.

10. Panchenko T. V. Geneticheskiye algoritmy [Genetic algorithms]. Astrakhan: Astrakhan University, 2007. P. 87.