Гибридные генетические алгоритмы для решения задач условной оптимизации

А. Е. Заварухин*, М. С. Семишкин‡, Е. В. Хворостухина§

Описан гибридный генетический алгоритм для одномерной и многомерной задач о рюкзаке. Реализация на Rust сочетает параллельную островную модель, жадный ремонт хромосом и локальный поиск. Приведены результаты вычислительных экспериментов.

Ключевые слова: задача о рюкзаке, многомерная задача о рюкзаке, генетический алгоритм, параллельные вычисления.

1. Введение и постановка задачи

Задачи условной оптимизации в общем случае характеризуются высокой вычислительной сложностью. Наличие жестких ограничений существенно сужает область допустимых решений и затрудняет поиск глобального экстремума классическими детерминированными методами. В данных условиях эффективным подходом становится применение метаэвристических гибридных алгоритмов, сочетающих в себе глобальный эволюционный поиск со специализированными локальными эвристиками и операторами. В данной работе предлагается архитектура гибридного параллельного генетического алгоритма (далее ГПГА), предназначенного для решения задачи об одномерном и многомерном рюкзаке. Применение точных методов для ее сложных экземпляров оказывается чрезмерно ресурсоёмким [1, 2].

Рассмотрим задачу выбора n предметов с ценностями $p_i > 0$, $i = \overline{1,n}$. Требуется найти бинарный вектор $x = (x_1, \ldots, x_n)$, максимизирующий ценность при ресурсных ограничениях: $\max_{x\in\{0,1\}^n} \sum_{i=1}^{n} p_i x_i$,

* *Заварухин Андрей Евгеньевич* — магистрант МГТУ им. Н. Э. Баумана, e-mail: zavaruhin.andrew@yandex.ru.

Zavarukhin Andrej Evgenevich — Master's student, BMSTU.

‡ *Семишкин Максим Сергеевич* — аспирант каф. ИКСП ИнПИТ СГТУ им. Гагарина Ю. А., e-mail: semishkin@bk.ru.

Semishkin Maxim Sergeevich — PhD student, SSTU, Dep. of Inform. and Commun. Systems and Software Engineering.

§ *Хворостухина Екатерина Владимировна* — канд. физ.-мат. наук, доцент каф. ИКСП ИнПИТ СГТУ им. Гагарина Ю. А., e-mail: katyanew2007@rambler.ru, ORCID: 0000-0002-2775-5732.

Khvorostukhina Ekaterina Vladimirovna — PhD, Associate professor, SSTU, Dep. of Inform. and Commun. Systems and Software Engineering.

Опубликовано на условиях лицензии CC-BY 4.0
Published under CC-BY 4.0

$\sum_{i=1}^{n} w_{ji}x_i \le c_j$, $j = 1, \ldots, m$. Здесь $w_{ji} \ge 0$ — расход ресурса j предметом i, c_j — доступный объём. При $m = 1$ задача одномерная (КР), при $m > 1$ — многомерная (МКР).

2. Алгоритм и программная реализация

Генетический алгоритм (ГА) — эволюционный метод поиска, в котором особи (возможные решения) последовательно изменяются с помощью операторов отбора, скрещивания и мутации. Каждая особь моделируется одной хромосомой. В нашем случае хромосома представляет собой бинарный вектор x. Функция приспособленности определяется следующим образом: $F(x) = \sum_{i=1}^{n} p_i x_i$. Оператор отбора выбирает родителей. В предлагаемом алгоритме реализовано два типа отбора: рулеточный, отбирающий особи с вероятностью, зависящей от F, и турнирный, когда выбирают лучшую из k случайных особей (в экспериментах $k = 3$). Использовался одноточечный оператор скрещивания: новые потомки образуются с помощью обмена фрагментами хромосом в одной точке разрыва. Оператор мутации случайно инвертирует отдельные биты. В предлагаемой реализации вероятности скрещивания и мутации были подобраны на основе предварительного параметрического анализа и равны соответственно 0,8 и 0,05. Новое поколение формируется по схеме $(\mu + \lambda)$. Это означает, что из μ родителей создаются λ потомков, а потом выбираются μ лучших из объединения родителей и потомков, образуя новое поколение. В нашем случае $\mu = \lambda$. Данная схема селекции по сути реализует принцип элитизма, гарантируя сохранение наиболее приспособленных хромосом на каждой итерации алгоритма.

Дополнительно применяются два специализированных оператора — жадный ремонт и локальный поиск. Первый в одномерной задаче удаляет предметы по возрастанию отношения ценности к весу до выполнения ограничения (DROP), а затем заполняет свободное место предметами с наибольшим отношением (ADD). Для МКР применяется псевдоэффективность — отношение ценности предмета к нормализованному потреблению им ресурсов. Локальный поиск дорабатывает лучшие хромосомы заменами типов один на один, один на два или два на один, добавляя наиболее ценные невыбранные и удаляя наименее ценные выбранные предметы. Замена принимается только при улучшении значения F без нарушения ограничений. Число итераций локального поиска ограничено. С вероятностью 0,2 поиск применяется к двум лучшим особям каждого поколения. Такое сочетание эволюции и локального улучшения соответствует меметическому подходу [3].

Программная реализация на Rust использует паттерн Builder и трейты для взаимозаменяемости операторов. Единое эволюционное ядро поддерживает обе постановки задачи (КР и МКР). Для МКР веса задаются векторами, а допустимость проверяется по всей системе ограничений. Интерфейс на базе `eframe` позволяет настраивать ГПГА и визуализировать графики лучшего значения F по поколениям для оценки скорости сходимости и выявления стагнации. Параллелизм реализован на базе островной модели [3]: субпопуляции эволюционируют в независимых потоках Rust с периодическим обменом особями. В экспериментах задействованы 4 острова (субпопуляция — 500 особей) с полносвязной топологией, миграция — 10% особей каждые 100 поколений с заменой худших хромосом на мигрантов, общее число поколений — 500.

3. Результаты вычислительных экспериментов

Для КР тестирование проводилось на бенчмарках Д. Писинжера, для МКР — на наборах Чу и Бизли [2] (по 20 запусков). Часть полученных данных сведена в общую табл. 1, где F^* — известный оптимум, $\overline{F}$ — среднее значение F, $F_{\max}$ — наибольшее F, $\delta = 100(F^* - \overline{F})/F^*$ (%) — среднее отклонение, $\bar{t}$ — среднее время одного запуска в секундах.

Таблица 1: Результаты экспериментов для ГПГА КР и МКР

Одномерная задача (КР)					Многомерная задача (МКР, $n = 500$)					
n	F^*	$\overline{F}$	$F_{\max}$	$\bar{t}$, с	m	F^*	$\overline{F}$	$F_{\max}$	δ, %	$\bar{t}$, с
1000	54 503	54 503	54 503	1,45	5	120 148	120 011	120 080	0,11	1,09
5000	276 457	276 456	276 457	7,94	10	117 726	117 281	117 498	0,38	2,52
10000	90 204	90 203	90 204	14,89	30	300 536	299 702	300 322	0,28	7,32

В одномерном случае во всех трех задачах $\delta < 0{,}0012\%$, поэтому в табл. 1 δ отсутствует, в остальных тестах для КР $\delta \leq 0{,}042\%$. При $n = 1000$ все 10 запусков достигли F^*. Для сравнения: ГА без специализированных операторов на тех же экземплярах имел отклонение 1,7–36,9%. Снижение ошибки сопровождалось ожидаемым ростом времени вычислений (при $n = 5000$ — с 4,01 до 7,94 с). Точные методы MinKnap и Combo справлялись с этими задачами быстрее, однако, в отличие от разработанного ГПГА, не смогли найти допустимые решения на экземплярах с большим диапазоном значений весов и стоимостей (10^7). На семи конфигурациях МКР ($n = 100$–500, $m = 5, 10, 30$) отклонения составили 0,11–0,71%. Характерно, что для сложного экземпляра $(n, m) = (500, 30)$ солвер IBM ILOG CPLEX (на платформе NEOS [4]) исчерпал лимит памяти через 10 203 с работы, так и не доказав оптимальность решения

300 536. ГППГА нашел сопоставимое по точности решение ($F_{\max} = 300\,322$) всего за 7,32 секунды. Таким образом, предложенный метод обеспечивает нахождение квазиоптимальных решений на рассмотренных инстансах (отклонение до 0,042% для КР и 0,71% для МКР) за приемлемое время.

Список литературы

[1] D. Pisinger, "Where are the hard knapsack problems?", *Computers & Operations Research*, **32**:9 (2005), 2271–2284. DOI: 10.1016/j.cor.2004.03.002.

[2] P. C. Chu, J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem", *Journal of Heuristics*, **4**:1 (1998), 63–86. DOI: 10.1023/A:1009642405419.

[3] *Parallel Metaheuristics: A New Class of Algorithms*, eds. E. Alba, John Wiley & Sons, Hoboken, NJ, 2005. DOI: 10.1002/0471739383.

[4] NEOS Server, *CPLEX solver*, https://neos-server.org/neos/solvers/lp:CPLEX/LP.html .

Статья поступила 23 октября 2026 г.

Hybrid Genetic Algorithms for Constrained Optimization Problems

A. E. Zavarukhin, M. S. Semishkin, E. V. Khvorostukhina

We present a hybrid genetic algorithm for one- and multidimensional knapsack problems. Its Rust implementation combines parallel island evolution, greedy chromosome repair, and local search.

Keywords: knapsack problem, multidimensional knapsack problem, genetic algorithm, parallel computing.

References

[1] D. Pisinger, "Where are the hard knapsack problems?", *Computers & Operations Research*, **32**:9 (2005), 2271–2284. DOI: 10.1016/j.cor.2004.03.002.

[2] P. C. Chu, J. E. Beasley, "A genetic algorithm for the multidimensional knapsack problem", *Journal of Heuristics*, **4** (1998), 63–86. DOI: 10.1023/A:1009642405419.

[3] *Parallel Metaheuristics: A New Class of Algorithms*, eds. E. Alba, John Wiley & Sons, Hoboken, NJ, 2005. DOI: 10.1002/0471739383.

[4] NEOS Server, *CPLEX solver*, https://neos-server.org/neos/solvers/lp:CPLEX/LP.html.

Received on October 23, 2026