

ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ СОСТАВЛЕНИЯ РАСПИСАНИЯ ПРОХОЖДЕНИЯ СУДОВ ЧЕРЕЗ ШЛЮЗОВАННЫЕ СИСТЕМЫ

USE OF PARALLEL GENETIC ALGORITHMS FOR SCHEDULING OF SHIPS PASSING THROUGH THE GATEWAYS SYSTEMS

В статье предлагается метод автоматического составления расписания прохождения судов через шлюзованные системы, основанный на использовании генетических алгоритмов и распределенных вычислений.

In this article a method for automatically scheduling the passage of ships through the gateways system based on genetic algorithms and distributed computing is proposed.

Ключевые слова: генетические алгоритмы, составление расписания, параллельные вычисления.

Key words: genetic algorithms, scheduling, distributed computing.

Введение

В современной России эксплуатируются десятки тысяч километров внутренних водных путей, на которых функционирует более 150 шлюзов. Эффективность эксплуатации флота на водных путях в значительной степени зависит от времени прохождения через судоходные гидротехнические сооружения, а также от стабильности и надежности работы судоходных шлюзов.

Известно, что шлюзование является одним из наиболее эффективных способов улучшения судоходных условий. Однако опыт эксплуатации показывает, что шлюзование имеет и негативную сторону: судоходные шлюзы являются своего рода тормозом транспортного процесса. Более того, вследствие характерной для речного флота неравномерности распределения грузо- и пассажиропотоков (как в течение суток, так и во время всей навигации) нередко происходит скопление в бьефах судов и составов, ожидающих шлюзования. Очевидно, что от качества расписания шлюзований зависит эффективность эксплуатации шлюзованной системы. Поэтому составление оптимального расписания шлюзований является важной задачей для повышения эффективности эксплуатации водного транспорта.

Применение

Постановка задачи

Рассмотрим задачу составления расписания прохождения судов через шлюзованные системы. Пусть шлюзованная система (ШС) двухниточная, состоит из однокамерных шлюзов (на примере шлюзов № 7 и 8 Шекснинского гидроузла); время подхода к подходному каналу ШС и состав групп судов, проходящих через данную ШС, известны заранее; также известно минимальное среднее время шлюзования. Тогда требуется составить расписание прохождения групп судов через данную ШС (план-график шлюзований) таким образом, чтобы все группы судов были включены в расписание; не возникало накладок в расписании; время ожидания шлюзования групп судов в подходном канале было минимальным; время простоя ШС было минимальным; соблюдалась последовательность пропуска судов, определенная в Правилах пропуска судов и составов через шлюзы внутренних водных путей РФ [1].

Математическая модель

В качестве исходных данных для составления расписания выступают множества:

1. $G = \{g_1, g_2, g_3, \dots, g_{N_g}\}$ нитей шлюза;
2. $S = \{s_1, s_2, s_3, \dots, s_{N_s}\}$ судов;
3. $T = \{t_1, t_2, t_3, \dots, t_{N_t}\}$ временных интервалов, описывающее режим работы ШС;

где N_g — количество нитей шлюза, N_s — количество судов, N_t — количество временных интервалов.

Теоретико-множественное описание элементов математической модели

Объект «Группа судов»

Согласно правилам [1] с целью наиболее эффективного использования судоходных шлюзов и транспортного флота, а также экономии воды разрешается совместное шлюзование судов. При составлении вариантов групп судов необходимо придерживаться рекомендаций [2].

Данный объект описывает свойства группы судов, такие как:

- направление движения группы судов;
- время подхода группы судов к подходному каналу ШС;
- время шлюзования группы судов через данную ШС;
- максимальное время ожидания шлюзования данной группы судов.

Тогда теоретико-множественное описание объекта «Группа судов» можно представить следующим образом:

$$S = \{s_i\}, s_i = (s_a^i, s_v^i, s_t^i, s_p^i, s_d^i, s_t^i), \quad (1)$$

где s_i — i -й элемент множества групп судов $S(i = \overline{1, N_s})$; s_a^i — время подхода группы судов к подходному каналу ШС (из множества T) $s_a^i \in T$; s_v^i — максимальное время ожидания шлюзования данной группой судов (из множества T) $s_v^i \in T$; s_t^i — время шлюзования группы судов через данную ШС (количество временных промежутков $t_{\min}$, необходимое для шлюзования); $s_d^i \in (0, 1)$ — направление движения группы судов через шлюз: если $s_d^i = 0$, то группа судов движется вверх по течению, иначе вниз по течению; s_t^i — параметр, определяющий приоритет данной группы судов в очереди на прохождение ШС.

Рассмотрим параметр s_t^i подробнее. Очередность прохода судов через ШС определяется в зависимости от типа судна. Согласно [1] можно выделить 6 типов судов, расположенных по убыванию приоритета прохода через ШС:

- 1) суда, имеющие право внеочередного прохода через ШС;
- 2) суда, следующие по расписанию;
- 3) суда первой очереди;
- 4) суда второй очереди;
- 5) суда третьей очереди;
- 6) суда четвертой очереди.

Тогда параметр s_t^i может принимать значения $\{1, 2, \dots, 6\}$ и показывает приоритет данной группы судов (1 — наивысший приоритет, а 6 — наименьший).

Объект «Временной интервал»

Временной интервал, на который составляется расписание прохождения судов через ШС, разобьем на равные промежутки времени, равные среднему минимальному времени шлюзования $t_{\min}$. Тогда можно определить время, необходимое для шлюзования группы судов s_i , $s_t^i = k \cdot t_{\min}$, где k — количество временных промежутков, необходимое для шлюзования группы судов. Общее количество временных промежутков $N_t = \frac{t}{t_{\min}}$, где t — расчетное время работы шлюза в сутки.

$$T = \{t_i\}, i = \overline{1, N_t}. \quad (2)$$

При составлении расписания будем оперировать следующими множествами объектов:

- 1) $G = \{g_i\}, i = 1, \dots, N_g$ — множество нитей шлюза;
- 2) $S = \{s_1, s_2, s_3, \dots, s_{N_s}\}$ — множество судов (групп судов);
- 3) $T = \{t_j\}, j = 1, \dots, N_t$ — множество временных интервалов.

С учетом (1)–(2) расписание прохождения групп судов через ШС можно определить векторами (γ, τ) :

$$\gamma = (\gamma_1, \dots, \gamma_i, \dots, \gamma_{N_g}), \quad \tau = (\tau_1, \dots, \tau_j, \dots, \tau_{N_t}), \quad (3)$$

где $\gamma_i \in G$ — код нити шлюза, назначенной группе судов для прохождения ШС; $\tau_j \in T$ — код временного интервала, назначенного группе судов для прохождения ШС.

Описание ограничений

Все требования, предъявляемые к расписанию прохождения судов через ШС, разобьем на обязательные и желательные. К обязательным относятся требования, невыполнение которых делает невозможным осуществление процесса пропуска судов через ШС. Эти требования будем рассматривать в качестве ограничений. К желательным относятся, например, минимизация времени стоянки судов в ожидании шлюзования и т. д. Эти требования будем рассматривать в качестве критериев.

Отсутствие накладок шлюзования групп судов

$$\forall (g_i, t_k) : g_i \in G, t_j \in T \left(\exists ! g_r : (g_i = g_r) \wedge (g_i \in Z^{t_j}) \right) \vee \left(\neg \exists g_r : (g_i = g_r) \wedge (g_i \in Z^{t_j}) \right), \quad (4)$$

где Z^{t_j} — множество шлюзований, происходящих во временной интервал t_j .

Критерий качества расписания

Для оценки полученного расписания введем критерий P качества расписания:

$$P = f(\tau) = \sum_{i=1}^N c_i w_i(\tau), \quad (5)$$

где c_i — коэффициент штрафа за невыполнение i -го требования, w_i — оценка, определяющая степень выполнения i -го требования. Чем больше значение критерия P , тем хуже расписание.

Теперь сформулируем задачу составления расписания. Для заданных множеств (1)–(2) требуется найти такое расписание, определенное вектором (3), содержащим коды временных интервалов, назначенных группам судов, которое удовлетворяет ограничениям и минимизирует значение критерия (5) потерь качества расписания:

$$P \rightarrow \min. \quad (6)$$

Алгоритм составления расписания

Задача составления расписания прохождения судов через шлюзованные системы относится к классу NP-полных задач, сложность решения которых растет экспоненциально с ростом числа и возможных значений изменяемых переменных. Поэтому для решения таких задач целесообразно использовать эвристические методы. Одними из самых популярных и перспективных эвристических алгоритмов на сегодняшний день являются генетические алгоритмы.

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путем случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. Является разновидностью эволюционных вычислений, с помощью которых решаются оптимизационные задачи с использованием методов естественной эволюции, таких как наследование, мутации, отбор и скрещивание.

Для решения задачи составления расписания прохождения групп судов через ШС применим классический генетический алгоритм (рис. 1), состоящий из следующих шагов:

- 1) создать начальную популяцию (множество допустимых расписаний);
- 2) принять начальную популяцию как текущее поколение;
- 3) отобрать из текущего поколения наиболее приспособленных особей в промежуточное поколение;
- 4) произвести над ними операции скрещивания и мутации;
- 5) создать путем отбора наиболее приспособленных из получившихся особей новое поколение;
- 6) если не выполнен критерий останова, то принять новое поколение как текущее и перейти к п. 3, иначе принять лучшую особь как окончательное решение.

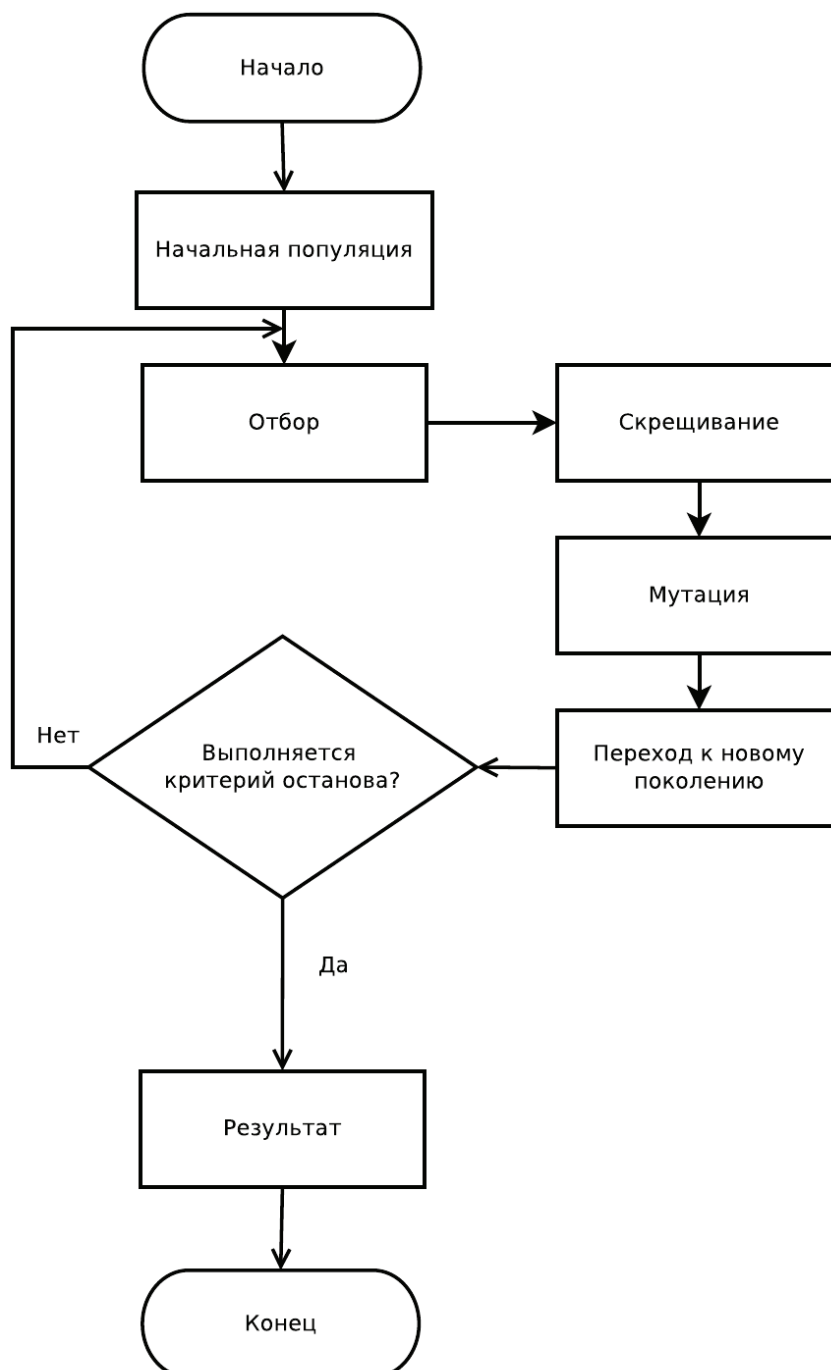


Рис. 1. Схема работы классического генетического алгоритма

Структура особи

В генетическом алгоритме решения задачи составления расписания прохождения групп судов через ШС каждая особь является одним из возможных решений задачи, то есть вариантом расписания. Формула (3), как было сказано, выражает математическую модель расписания. Следуя этой модели, предлагается рассматривать особь, состоящую из одной хромосомы (рис. 2).

Хромосома, в свою очередь, состоит из генов, обозначаемых целыми числами $1, 2, 3, \dots, i, \dots, N$, причем номер гена соответствует номеру группы судов, так, i -й ген характеризует группу судов g_i из множества G .

Информационным наполнением хромосомы является время прохода группы судов через ШС. Таким образом, значением i -го гена является номер (код) временного интервала из подмножества допустимых временных интервалов, в который предполагается начать проведение группы судов через ШС. Это означает, что хромосома связана с блоком групп судов особой связью, которую можно назвать связью «однозначного соответствия».

Создание начальной популяции

Случайным образом формируется заданное число особей. Формирование каждой особи происходит следующим образом. Для каждого гена хромосомы, условно обозначающего группу судов, приписывается некоторое случайное значение — номер временного интервала из подмножества допустимых для данной группы судов временных интервалов. Аналогичным образом формируются следующие особи популяции.

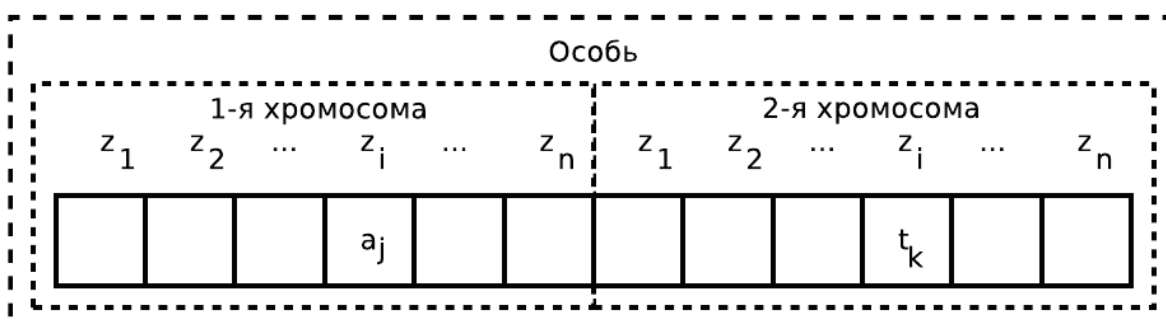


Рис. 2. Структура особи

Отбор

На данном шаге происходит отбор наиболее приспособленных особей (вариантов расписания), имеющих наиболее предпочтительные значения используемой оценочной функции по сравнению с остальными особями. В данной работе предлагается метод, называемый «элитным отбором» или «элитной стратегией», который при решении данной задачи заключается в следующем: из предыдущей популяции выбирается только некоторое число отдельных особей, имеющих наименьшее значение весовой функции — критерия, отражающего выполнение желательных требований. Выявленные таким способом «элитные» особи без каких-либо изменений переходят в следующее поколение. Оставшееся количество «свободных мест» в новой популяции заполняется особями, полученными в результате их скрещивания и мутации (рис. 3).

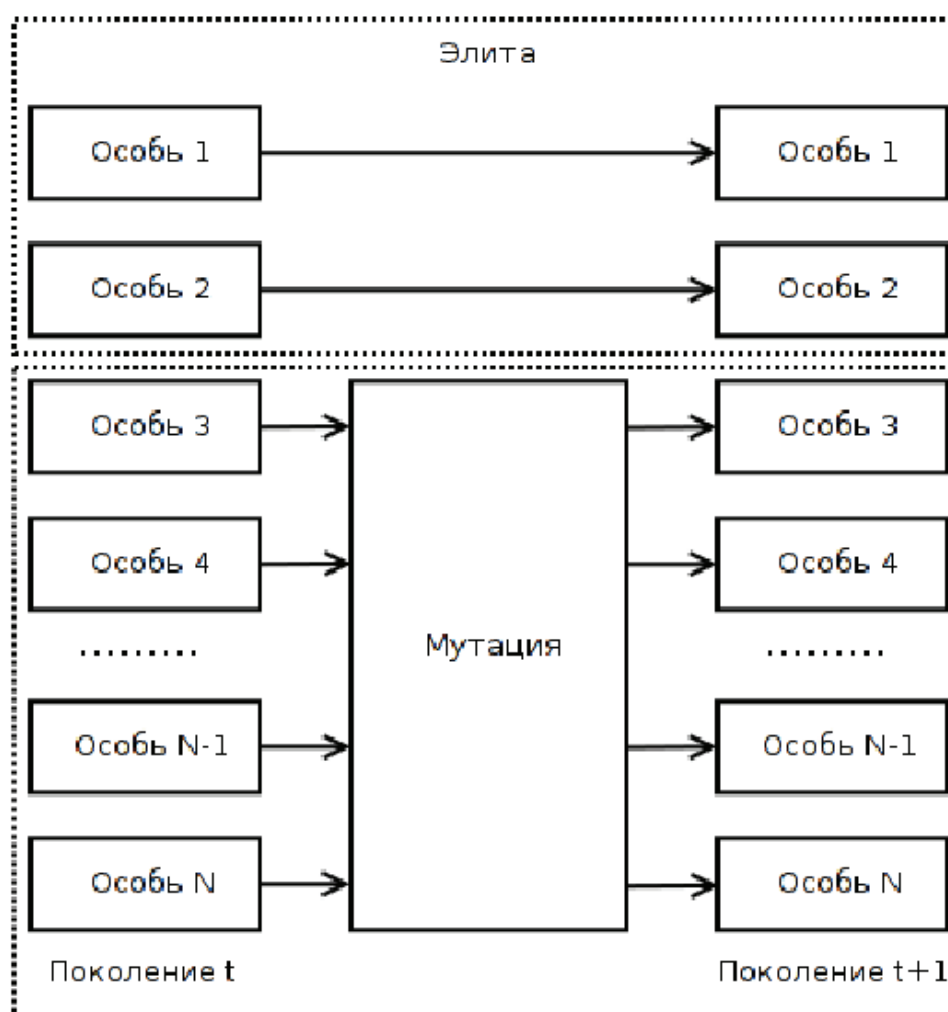


Рис. 3. Схема отбора

Скращивание

В предлагаемом генетическом алгоритме используется одноточечное скрещивание. Оно происходит по следующей схеме: случайным образом выбираются n пар особей из текущего поколения. Далее для каждой пары отобранных особей случайным образом разыгрывается позиция гена (локус) и производится обмен участками генетического кода между особями (рис. 4). Таким образом получается новая пара особей, которая помещается в новое поколение.

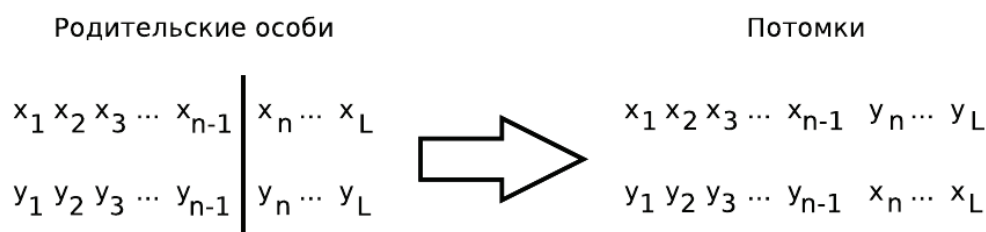


Рис. 4. Скрещивание

Мутация

При формировании нового поколения к особям предыдущего поколения, не относящимся к «элитным», применяется операция мутации.

В данной задаче реализованы два вида мутации:

1) случайным образом выбирается одна группа судов. Если для нее доступны другие временные интервалы, в которые не нарушается ее расписание, то шлюзование данной группы судов переносится на один из этих интервалов, выбранный случайным образом;

2) случайным образом выбираются два шлюзования, которые проводятся для судов одного типа. Если возможно, то эти шлюзования меняются местами.

Выбор вида мутации производится случайным образом.

Критерий останова и выбор «наилучшей» особи

В результате применения операторов отбора и мутации формируется популяция потомков, которая заменяет родительскую популяцию, после чего выполняется проверка условия останова алгоритма. Если заданное значение максимального количества итераций алгоритма достигнуто, то алгоритм завершает работу. В последнем поколении в качестве решения задачи выбирается особь, которая в наибольшей степени удовлетворяет требованиям, предъявляемым к расписанию. То есть особь, имеющая минимальное значение оценочной функции.

Применение параллельных вычислений для решения задачи

Известно, что генетические алгоритмы хорошо поддаются распараллеливанию. Для них предложено множество архитектур и стратегий поиска для выполнения на параллельных процессорах. Данные стратегии основаны на взаимодействии изолированных популяций или отдельных генетических алгоритмов, вычисление решений которых реализуется на отдельных процессорах. Взаимодействие популяций или генетических алгоритмов реализуется по специальным правилам, формирующим суть стратегии и требующим специальной параллельной архитектуры процессоров. Однако в настоящий момент широко распространены и общедоступны многоядерные процессоры (2, 3 и 4-ядерные), где каждое ядро представляет собой полноценный процессор. Данную систему можно рассматривать как простейшую многопроцессорную систему. Рассмотрим несколько подходов к распараллеливанию генетических алгоритмов.

Первым подходом является распараллеливание отдельных шагов алгоритма: отбора, скрещивания, мутации, оценки особей. При этом генофонд разделяется на блоки, над каждым из которых работает отдельный поток. Однако такой подход не очень удобен в программировании по сравнению с двумя другими.

При втором подходе, его еще называют островной моделью, заводят несколько независимых генофондов (островов), каждый из которых итерируют в отдельном потоке. Затем каждые несколько итераций устраивают «миграцию» — переселение ДНК между всеми «островами». Стратегию переселения можно выбирать свободно: можно переселять чемпиона, можно случайную ДНК (или несколько ДНК). Однако стоит помнить, что слишком частые миграции, а также миграции большого числа ДНК нивелируют со временем популяции на разных «островах».

Такой подход имеет неоспоримые преимущества. Во-первых, его легко воплотить программно, так как не требуются затраты на синхронизацию потоков. Во-вторых, на каждом «острове» можно использовать различные стратегии мутации, размножения и т. д., что приведет к более широкому охвату области возможных значений переменных и с большей вероятностью к глобальному экстремуму, нежели локальному.

Третий подход является модификацией островной модели из расчета на массовый параллелизм. Его называют клеточной моделью. Такой подход можно использовать, когда речь идет о вычислениях в кластере. Суть клеточной модели в следующем: пусть у нас имеется множество процессоров (машин), расположим их логически в виде таблицы.

Каждая ячейка таблицы представляет собой один процессор. Каждый процессор итерирует отдельную популяцию со своим набором стратегий размножения, мутации и т. д. По окончании определенного числа итераций происходят миграции между соседними ячейками таблицы.

Такой подход применим лишь на многоядерных машинах либо в кластерах, так как на машинах с малым числом ядер возникнут большие накладные расходы, связанные с переключением контекста потока, что резко снизит эффективность параллельных вычислений.

Преимуществом клеточной модели перед островной является резкое снижение вероятности скатывания в локальный минимум, так как перемешивание происходит не между всеми популяциями, как в островной модели, а лишь между соседними, что снижает скорость выравнивания генофонда между популяциями.

Для решения данной задачи был использован первый подход к реализации параллельного генетического алгоритма — распараллелить процесс выполнения его операторов и вычисление значения оценочной функции. В случае параллельных операторов алгоритм будет работать следующим образом. Базовый поток порождает рабочие потоки, которые могут обмениваться сообщениями с ним с помощью некоторого интерфейса. Когда вычисления в базовом потоке доходят до параллельного оператора, формируются задания рабочим потокам. Популяция делится на равные части (по количеству рабочих потоков), и каждая часть посылается соответствующему рабочему потоку. Базовый поток продолжает свою работу после получения результатов от всех рабочих потоков. Если в процессе ожидания результатов делается вывод о некорректном завершении какого-либо рабочего потока, задание посылается повторно другому рабочему потоку, свободному в данный момент. Алгоритм работы многопоточной программы, реализующий данную идею, показан на рис. 5.

Важно заметить, что при такой организации ГА ему безразлично, как и в какой среде передаются сообщения, поэтому этот способ подходит для вычислительных систем любой организации: однопроцессорных ЭВМ, многопроцессорных ЭВМ или кластеров любой структуры. Эффективность такого распараллеливания зависит прежде всего от скорости передачи сообщений между потоками. Если рассматривать многопроцессорную ЭВМ с общей памятью, где скорость передачи данных между потоками максимальна, то распараллеливание любых операторов эффективно практически во всех случаях, так как генотипы, хранящиеся в общей памяти, не копируются для каждого рабочего потока, а используются совместно.

Если же рассматривать кластер, где оперативная память у потоков разделена, а передача сообщений производится через локальную сеть, то необходимо произвести детальный анализ последствий распараллеливания того или иного оператора с точки зрения потерь производительности алгоритма при передаче сообщений между потоками. В целом оператор целесообразно распараллеливать, если вычисление результата для каждого генотипа занимает гораздо большее время, чем передача генотипа в сообщении потоку и прием ответа.

Инициализация генетического алгоритма и запуск дополнительных потоков происходят из главного потока, который создается при запуске программы. Количество запускаемых потоков равно количеству ядер процессора. Как видно из рис. 5, дополнительные потоки не изменяют структуру генетического алгоритма, а только дополняют ее. Программа может автоматически запрашивать у системы количество процессорных ядер и запускать, если это возможно, дополнительные потоки.

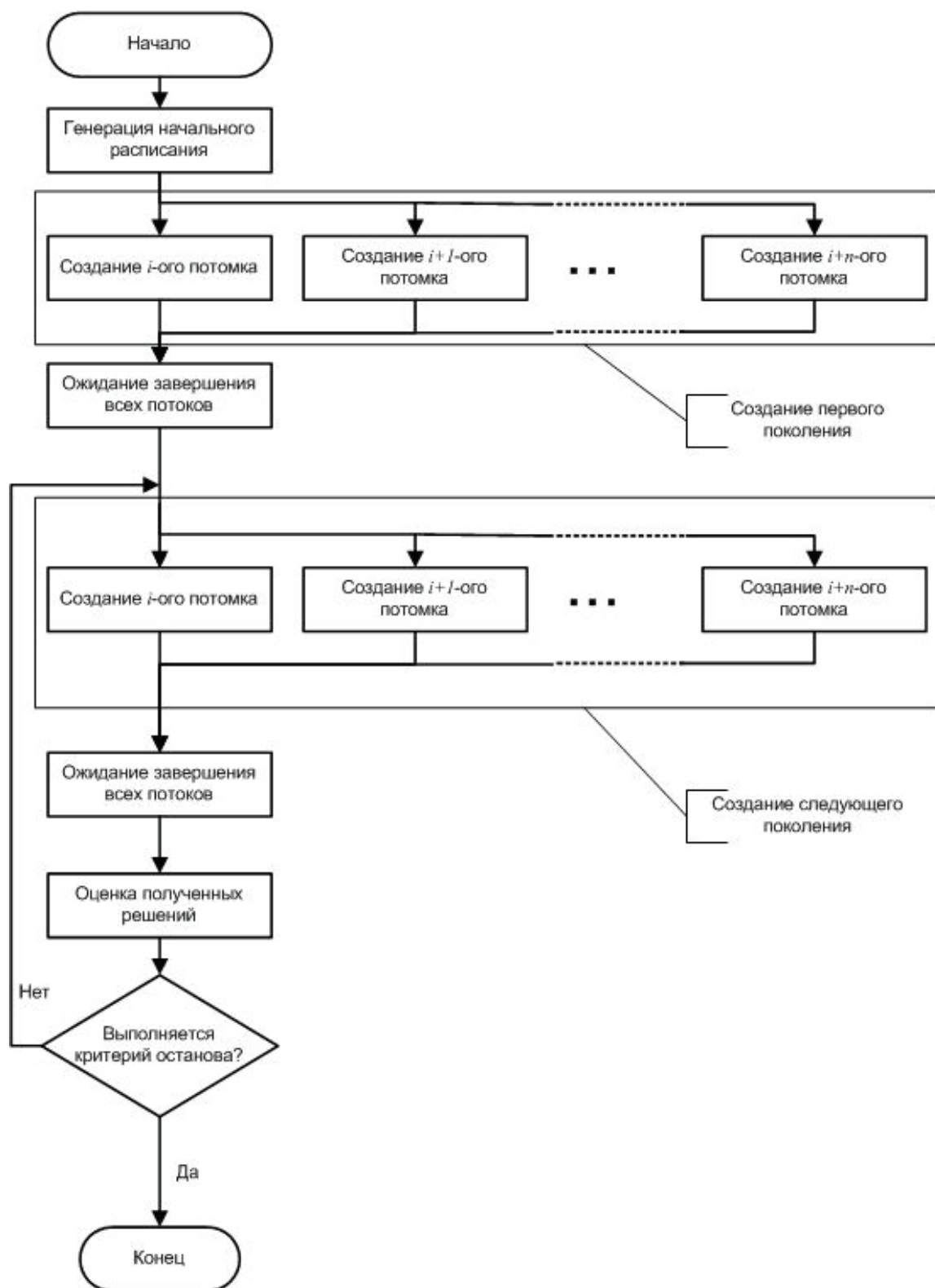


Рис. 5. Алгоритм программы формирования популяции

Широко известно, что в многопоточной программе работа потоков тем эффективнее, чем с меньшим количеством общих изменяемых данных работают потоки. Это связано с тем, что если потоки изменяют какие-то общие данные, то необходимо применять специальные дополнительные механизмы межпоточной синхронизации (критические секции, семафоры, события и др.).

В данном случае потоки работают независимо друг от друга. И лишь при завершении работы поток записывает полученные данные в общую память. Тем самым достигается высокая скорость работы программы.

Заключение

Предложенный алгоритм может быть использован для составления расписания прохождения судов через отдельную шлюзованную систему. Среди преимуществ можно отметить возможность генерации приемлемых вариантов расписания уже с первой итерации. В алгоритме предусмотрена возможность значительного улучшения расписания благодаря добавлению дополнительных критериев оценки свободы и качества расположения групп судов в расписании.

Список литературы

1. Правила пропуска судов и составов через шлюзы внутренних водных путей Российской Федерации: утв. приказом Минтранса РФ от 24 июля 2002 г. № 100.
2. Кононов В. В. Гидротехнические сооружения водных путей, портов и континентального шельфа (судоходный канал и бетонный шлюз) / В. В. Кононов. — СПб.: СПГУВК, 2009.