Ответы на замечания по статье
«ПРИМЕНЕНИЕ МЕТОДА ПРОЕКТИРОВАНИЯ Q-ЭФФЕКТИВНЫХ ПРОГРАММ ДЛЯ АЛГОРИТМА ДЕЙКСТРЫ»

Авторы выражают благодарность рецензентам за ценные замечания, способствующие улучшению текста статьи.

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
	Рецензент А	
	Содержание	
В ответах на замечания рецензента авторы зачастую приводят обоснованные аргументы, не вполне отраженные, к сожалению, в исправленном тексте статьи. В этой связи отмечу, что целью рецензента является, прежде всего, улучшение текста статьи, а затем -- постижение резонов, по которым авторы отказываются исправлять высказанные замечания.		
1.	По ответу на замечание 2.3. Авторы обосновывают проведенные ими эксперименты (далее цитаты из текста статьи: "для разных графов с одинаковым числом вершин", где "в качестве результата" бралось среднее арифметическое времени выполнения для исследования эффективности своего решения "в среднем случае"), ссылаясь на проведенный ими анализ полученных рядов измерений времени выполнения для определенных сочетаний конфигурации и размерности задачи и полученный ими факт подчинения значения времени выполнения закону нормального распределения. Однако в тексте статьи указанный анализ и его результаты не упомянуты.	Выполнено. Добавлен краткий анализ, обосновывающий использование среднего арифметического для нескольких повторений эксперимента.
2.	По ответу на замечания 3, 3.2. Авторы обосновывают отсутствие сравнения своего решения с аналогами. Следует добавить соответствующее краткое пояснение в текст статьи. В данном пояснении совершенно уместно будет процитировать все ранее рекомендованные рецензентом аналоги (иные решения по распараллеливанию алгоритма Дейкстры). В исправленной версии текста аналоги [14, 18] цитируются в некорректном существенно общем контексте как "параллельные программы, учитывающие специфику алгоритмов и архитектуры ПВС".	Выполнено. Краткое пояснение обоснования отсутствия сравнения решения авторов с аналогами добавлено в конец раздела 1 статьи. Все три ранее рекомендованных рецензентом аналога (иные решения по распараллеливанию алгоритма Дейкстры) включены в список литературы и процитированы.
3.	По ответу на замечания 4.1, 4.2. Авторы вновь приводят некую аргументацию в своем ответе, но не отражают ее в исправленном тексте статьи (в т.ч. ссылаясь в ответе на источник, не цитируемый в статье). Тем не менее, считаю, что проблема с подачей материала остается. По-прежнему считаю используемое авторами определение ускорения неточным, а контекст его применения -- запутывающим читателя. Авторы ссылаются на определение этого	Выполнено. Уточнено определение ускорения. Графики на рисунках были изменены. Теперь осью абсцисс является количество нитей или узлов. По замечанию рецензента В вместо последовательной программы используется вариант параллельной программы для общей памяти, запускаемый только на одном ядре и на одной нити. В таком случае время выполнения оказалось не меньше, чем у ранее исследованной последовательной программы.

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
	термина в Глоссарии проекта Алговики, но оно более точное, чем у авторов, сравним: Авторы: Ускорение – отношение времени выполнения последовательной программы к времени выполнения параллельной программы. Алговики: Ускорение работы программы S=T_1/T_p – отношение времени работы некоторой программы на одном процессоре T_1 ко времени работы распараллеленной программы при использовании p процессоров T_p. Термин "ускорение", как мы видим, тесно связан с количеством исполняющих устройств, и здесь естественно ожидать отображения результатов исследования в виде графика, в котором осью абсцисс является количество этих устройств (а не объем данных). При этом для случая общей памяти в качестве p будет фигурировать количество нитей, для распределенной памяти -- количество узлов кластера (или процессоров, если узлы многопроцессорные). Кстати, при исследовании на общей памяти следует учитывать количество физических ядер и гипертрединг при назначении max количества нитей и указывать это в описании экспериментов. Соответственно, графики на рис. 1, 2 оставляют много неясностей и вопросов, указанных ранее. Попутно отмечу, что в графике эффективности распараллеливания на рис. 1в в легенде присутствует позиция "последовательная программа", смысл которой неясен. Если авторы хотят исследовать масштабируемость алгоритма в слабом смысле (или эквивалентный термин "расширяемость", используемый рецензентом), то это следует явно прояснить в описании целей экспериментов. В любом случае, мне представляется, что для авторов не будет проблемой привести как график(и) ускорения (в общепринятом смысле; зафиксировав один, несколько по своему выбору или все рассматриваемые графы), так и график расширяемости (в котором будут задействованы результаты экспериментов для всех рассматриваемых графов). Кроме того, не хотелось бы, чтобы создавалось ложное впечатление о том, что авторы исследуют только то ускорение	Последовательная программа в эксперименте больше не упоминается. Слабая масштабируемость в данной статье не исследуется и причины этого приведены в предыдущей версии рецензии. Напомним, что исследовать слабую масштабируемость не представляется возможным в полной мере, т.к. для полного исследования требуется 240 вычислительных узлов, а квота пользователя суммарно составляет не более 70 узлов.

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
	программы, которое "выгодно" покажет работу метода проектирования Q-эффективных программ, опуская при этом общепринятые. Здесь же отмечу, что авторы справедливо указали на опечатку в замечании рецензента при определении ускорения.	
4.	По ответу на замечание 5. Можно согласиться с тем, что авторы используют в экспериментах все доступные процессоры узла кластера. Но остается без ответа вопрос, зачем при этом на процессоре запускается меньшее количество нитей, чем это можно сделать на текущей аппаратной платформе. Это можно было бы понять, если бы авторы поступали так при исследовании ускорения (в общепринятом смысле) Q-эффективного алгоритма на общей памяти. Но неясно, зачем исследовать расширяемость Q-эффективного алгоритма на общей памяти, но на меньшем числе нитей. И тем более неясно, зачем исследовать ускорение/расширяемость Q-эффективного алгоритма на распределенной памяти, когда на каждом узле кластера запускается меньшее число нитей, чем это физически возможно.	Если отсутствует привязка нитей к ядрам и при запуске программы было указано, что задействуются все ядра процессора, то даже при количестве нитей меньшем, чем количество ядер, будет происходить следующее. Каждая нить будет постоянно переключаться между всеми ядрами. Тогда к концу выполнения программы существует вероятность события того, что для её выполнения были использованы все ядра процессора. Проверить действительность этого факта можно так. На любом дистрибутиве Linux нужно сделать следующее. В терминале запустить диспетчер задач top следующей командой: top -H -p <PID какого-нибудь процесса>. Нажать клавишу F, выбрать параметр Last Used CPU (P), нажать пробел, затем Esc, затем клавишу J. Появившийся параметр P указывает номер используемого ядра. Варьирование количества нитей позволяет выявить их влияние на изучаемые динамические характеристики. Важно отметить, в каждой части эксперимента присутствует конфигурация, где на каждом задействованном узле используется максимально возможное число нитей – 24. В наших экспериментах привязка нитей не осуществлялась, т.к. в версии планировщика SLURM, установленного на суперкомпьютере, такой функции нет. Насколько мне известно, в новых версиях SLURM такая привязка возможна. Обоснуем варьирование количества нитей в экспериментах. Целью было выявить наличие и влияние факторов, создающих накладные расходы при планировании нитей операционной системой (например, дисбаланс загрузки ядер). Данный подход не позволил выявить факторы, которые зависят от числа нитей, т.к. большая часть значений ускорения находится в узком диапазоне (1,7–1,8). Примечание. Еще раз обратим внимание, что нити – это абстракция операционной системы. Нити не являются чем-то материальным (в

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
		отличие от ядер процессора) и не могут быть «запущены физически».
5.	Переработка результатов экспериментов, очевидно, потребует изменения текста в части обсуждения результатов. В этой связи, думаю, читателю было бы естественным ожидать от авторов выводов, обсуждающих смысловую корреляцию понятий Q-эффективной программы и ускорения программы. В частности, насколько близким к линейному является ускорение Q-эффективной программы, от чего это зависит и др.	Выполнено. Текст, характеризующий выводы о результатах экспериментов, изменен.
	Оформление	
1.	В списках литературы для публикаций в трудах конференций [3, 15-17, 19-21, 24, 25] необходимо указать полное официальное название конференции, даты и место ее проведения (см. примеры оформления на сайте журнала: https://vestnik.susu.ru/cmi/pages/view/about-author-guidelines#format, пп. 26, 35).	Выполнено.
2.	Рис. 1, 2 не следует разрывать между страницами.	Выполнено. Каждый из рисунков 1 и 2 был разделен на несколько рисунков.
	Рецензент B	
1.	Про последовательную программу, с которой производится сравнение, авторами сказано только: "была разработана последовательная программа. Она представляет собой программу, выполняющую на одном вычислительном узле ту же задачу, которая сформулирована при проектировании Q-эффективной программы, и не использующую программные средства распараллеливания или межпроцессное взаимодействие." Эта формулировка не говорит ничего про эффективность реализации последовательного алгоритма, в то время как сравнение с заведомо плохим вариантом может использоваться для получения большего ускорения. Поэтому принято сравнивать с лучшим последовательным вариантом (но тогда следует обосновать, что он лучший), либо же с тем же параллельным вариантом, но запущенным на одном исполнительном устройстве (ядре процессора).	Выполнено. Было изменено определение ускорения. В связи с этим была повторно проведена часть эксперимента, затрагивающая параллельную программу для общей памяти. Эксперимент проводился на одном ядре процессора, использующего одну нить. Полученные результаты близки по значениям с предыдущими измерениями для последовательной программы.
2.	Не могу согласиться с тем, что для определения эффективности распараллеливания авторы используют только число процессоров – это было бы верно для программы только на MPI (без использования OpenMP). Если же используется OpenMP, то	Выполнено. Было изменено определение ускорения. Графики были изменены с учетом замечаний.

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
	корректно определять эффективность распараллеливания в зависимости от общего числа нитей. Получаемые авторами по своей методике значения эффективности распараллеливания для программы на общей памяти (рис.1) в пределах 0.7-0.9 должны означать крайне эффективную реализацию, что никак не соответствует ускорению всего в 1.3-1.8 раза, что вводит в заблуждение читателей. С эффективностью распараллеливания программы над распределенной памятью разобраться труднее (из-за большего разброса значений и сложности сопоставления трёхмерных графиков), но там наверняка подобная же картина.	
3.	Непонятно неоднократно повторяемое авторами утверждение, что "в экспериментах используются все процессоры задействованного узла и их ядра". Если запускается разное количество нитей на узел (как это тоже неоднократно описывается в статье), то часть ядер процессора в каких-то экспериментах остается незадействованными.	Если в данном пункте понимается, что конкретные ядра процессора **никогда ни при каких условиях** не будут использоваться для выполнения программы на всем протяжении времени выполнения, то это утверждение неверно в случае, если не используется привязка нитей к ядрам или другая подобная технология. Если при запуске программы было указано, что задействуются все ядра процессора (как в нашем случае), то даже при количестве нитей меньшем, чем количество ядер, будет происходить следующее. Каждая нить будет постоянно переключаться между всеми ядрами. Тогда к концу выполнения программы вероятно событие, что для её выполнения задействовались все ядра процессора. Описываемое явление не равнозначно тому, что определенная часть ядер просто остается незадействованной в выполнении процесса. Проверить действительность этого факта можно так. На любом дистрибутиве Linux нужно сделать следующее. В терминале запустить диспетчер задач top следующей командой: top -H -p <PID какого-нибудь процесса>. Нажать клавишу F, выбрать параметр Last Used CPU (P), нажать пробел, затем Esc, затем клавишу J. Появившийся параметр P указывает номер используемого ядра. В наших экспериментах привязка нитей не осуществлялась, т.к. в версии планировщика SLURM, установленного на суперкомпьютере, такой функции нет. Насколько мне известно, в новых версиях SLURM такая привязка возможна.

№ п/п	Замечание рецензента	Ответ авторов (отметка о выполнении и/или комментарий)
4.	В ответе авторы пишут: "Если воспользоваться законом Амдала, максимально достижимое значение ускорения 6 при работе 18 процессоров на 9 узлах соответствует доле параллельных вычислений около 85%, что однозначно указывает на высокую степень использования ресурса параллелизма алгоритма." На мой взгляд, для реальных вычислительных программ доля параллельных вычислений около 85% соответствует не высокой, а крайне низкой степени использования ресурса параллелизма. Если бы в реальных задачах доля последовательных вычислений составляла 15%, их распараллеливание почти всегда было бы невыгодно. Отсюда следует абсолютно противоположная оценка качества распараллеливания. Далее следует утверждение авторов: "Так как описанная Q-эффективная реализация использует весь ресурс параллелизма алгоритма, то полученное значение ускорения можно считать значением предельно достижимого ускорения в данной вычислительной системе." Если даже Q-эффективная реализация использует весь ресурс параллелизма, то это само по себе ничего не доказывает, потому что этот ресурс параллелизма может использоваться неэффективно.	По просьбе рецензента А авторы внесли дополнительные пояснения в статью. Они содержатся в разделе 1 в последнем абзаце. Авторы полагают, что эти пояснения вносят ясность по поводу вопросов, обсуждаемых в данном замечании.
5.	Оценка полученных результатов в конце раздела 4 по-прежнему является формальной и малосодержательной - авторы фактически описывают словами то, что они видят на приведенных выше графиках, но не приводят объяснений полученных результатов или их влияния на результаты работы.	Выполнено. Выводы были дополнены.