

ДВУХКРИТЕРИАЛЬНЫЙ АЛГОРИТМ ЭНЕРГОЭФФЕКТИВНОГО ПЛАНИРОВАНИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ В ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ ПОДВОДНОГО АППАРАТА

Е. Г. Литуненко

Концерн „ЦНИИ «Электроприбор»“, Санкт-Петербург, Россия
lisa.litunenko@gmail.com

Аннотация. Предложен алгоритм энергоэффективного планирования процессов в вычислительной системе подводного аппарата. Алгоритм обеспечивает минимизацию потребляемой вычислительной системой мощности и среднего времени пребывания заданий в системе. Отличительной особенностью алгоритма является то, что он позволяет осуществлять *flow shop*-планирование в системах с многими информационными выходами, которые могут появиться в результате построения энергоэффективной архитектуры системы. Использование алгоритма при проектировании макета вычислительной системы подводного аппарата позволило снизить потребляемую мощность и уменьшить среднее время пребывания заданий в системе, по сравнению с методом перебора в исходной системе. Полиномиальная сложность предложенного алгоритма позволяет использовать его при планировании процессов в системах реального времени.

Ключевые слова: планирование вычислений, распределенная вычислительная система, *flow shop*-планирование, энергоэффективные вычисления, навигационный комплекс, автономный необитаемый подводный аппарат

Ссылка для цитирования: Литуненко Е. Г. Двухкритериальный алгоритм энергоэффективного планирования информационных процессов в вычислительной системе подводного аппарата // Изв. вузов. Приборостроение. 2025. Т. 68, № 1. С. 13–22. DOI: 10.17586/0021-3454-2025-68-1-13-22.

TWO-CRITERIA ALGORITHM FOR ENERGY-EFFICIENT PLANNING OF INFORMATION PROCESSES IN THE COMPUTING SYSTEM OF AN UNDERWATER VEHICLE

E. G. Litunenko

Concern CSRI Elektropribor, JSC, St. Petersburg, Russia
lisa.litunenko@gmail.com

Abstract. An algorithm for energy-efficient planning of processes in the computing system of an underwater vehicle is proposed. The algorithm ensures minimization of the power consumed by the computing system and the average time tasks spend in the system. A distinctive feature of the algorithm is that it allows for *flow shop* planning in systems with many information outputs that may appear as a result of building an energy-efficient system architecture. Using the algorithm when designing a layout of the computing system of an underwater vehicle allows for reducing the power consumption and decreasing the average time of stay of tasks in the system compared to the enumeration method in the original system. The polynomial complexity of the proposed algorithm makes it possible to use it when planning processes in real-time systems.

Keywords: computing scheduling, real-time distributed computing system, power consumption reduction, *flow shop* scheduling, energy-efficient computing, autonomous underwater vehicle

For citation: Litunenko E. G. Two-criteria algorithm for energy-efficient planning of information processes in the computing system of an underwater vehicle. *Journal of Instrument Engineering*. 2025. Vol. 68, N 1. P. 13–22 (in Russian). DOI: 10.17586/0021-3454-2025-68-1-5-13-22.

Введение. В настоящее время ведутся активные работы по проектированию и исследованию автономных необитаемых подводных аппаратов (АНПА) в связи с широкими возможностями применения таких систем для исследований [1–3], поиска полезных ископаемых [4], патрулирования территорий и др. При проектировании АНПА необходимо учитывать высокие требования к автономности аппарата и как следствие — энергоэффективности входящих в него систем. АНПА зачастую снабжаются большим количеством навигационного оборудования,

вследствие чего на вычислительную систему ложится большая нагрузка по обработке навигационной информации. В связи с этим актуальной задачей является не только повышение энергоэффективности системы, но и снижение времени, необходимого для обработки информации. В этом случае важное значение приобретают алгоритмы планирования вычислений. Уместно отметить, что проблема планирования заданий является актуальной и в других отраслях и приложениях, среди которых планирование экономических и технологических процессов [5], планирование вычислений [6, 7], логистика [8] и др. При этом для разных приложений задачи планирования могут иметь схожую постановку, что позволяет применять одни и те же алгоритмы для планирования различных процессов.

В настоящей работе предлагается алгоритм планирования вычислений в распределенных вычислительных системах, а именно определения последовательности выполнения требуемого множества заданий. На подход к построению алгоритма существенно влияет выбор критерия оптимальности для формируемого плана. Примерами критериев могут служить минимум общего времени выполнения заданий, минимум максимального отклонения от директивных сроков [9]. Бортовые распределенные вычислительные системы, как правило, можно отнести к классу многоканальных систем с обработкой информации на общих процессорах. Подходящей моделью для них является *flow shop*-система, для которой разработаны эффективные алгоритмы планирования [5, 9, 10]. Однако известные алгоритмы *flow shop*-планирования нельзя применить к реальным системам напрямую. Это может быть связано, например, с отсутствием изоморфизма между графиками заданий и графиком системы. В работе [11] предложено преодолевать эту проблему с помощью добавления в графы заданий задач нулевой длительности. Совершенствование алгоритмов *flow shop*-планирования остается актуальным в связи с наличием и других препятствий, которые не позволяют использовать существующие алгоритмы.

Цель настоящей статьи состоит в разработке такого алгоритма планирования, который позволит снизить потребляемую системой мощность и среднее время пребывания заданий в системе. Особенность разрабатываемого алгоритма заключается в том, что он должен обеспечивать *flow shop*-планирование для систем с многими информационными выходами.

Концепция энергоэффективного подхода к планированию заданий в системе. В статье предлагается двухкритериальный алгоритм, который позволяет минимизировать потребляемую системой мощность и среднее время пребывания заданий в системе. Предлагаемый алгоритм является составным — сначала определяется энергоэффективная архитектура вычислительной системы с использованием известного алгоритма энергоэффективного назначения задач на процессоры [12], после чего осуществляется планирование заданий в преобразованной системе. Энергоэффективная архитектура A системы определяется в соответствии с алгоритмом, который предполагает с целью минимизации потребляемой мощности $W(A)$ снизить тактовую частоту и напряжение питания процессоров, т. е.

$$A^* = \arg \min_A W(A).$$

Предполагается, что исходное назначение заданий соответствует *flow shop*-системе. При этом особенностью получаемой архитектуры в общем случае является возникновение в системе многих информационных выходов, что не соответствует общепринятой постановке задачи при *flow shop*-планировании и делает известные алгоритмы планирования неработоспособными.

Алгоритм энергоэффективного назначения задач на процессоры. В основе первого шага предлагаемого двухкритериального алгоритма лежит определение энергоэффективной архитектуры системы на основе принципа, который изложим по отношению к системе, состоящей из одного процессора. Пусть тактовая частота и напряжение питания процессора снижены в k раз. Тогда в соответствии с представленными в работах [12, 13] зависимостями потребляемая мощность снизится в k^3 раз, а время работы процессора увеличится в k раз. Если для сохранения производительности процессора на прежнем уровне увеличить в k раз число процессоров,

а вычислительную нагрузку разделить между всеми процессорами поровну, то в результате потребляемая мощность уменьшится в k^2 раз по отношению к исходной.

Алгоритм энергоэффективного назначения задач на процессоры использует этот прием для каждого из N процессоров исходной системы, благодаря чему каждый процессор в общем случае замещается в формируемой архитектуре группой, обеспечивающей снижение энергопотребления. При этом предполагается, что:

- все процессоры исходной системы работают на одной частоте и при одном напряжении питания;
- для исследуемой системы заданы минимальные значения напряжения питания и частоты;
- ограничено число добавляемых в систему процессоров.

Более подробно этот алгоритм рассматривается в статье [14].

Алгоритмы построения частных планов системы flow shop с минимизацией среднего времени пребывания задания в системе. В общем случае в результате применения алгоритма энергоэффективного назначения задач на процессоры можно получить систему, архитектура которой имеет множество информационных выходов, известные алгоритмы *flow shop*-планирования для нее становятся неприменимыми. В связи с этим на следующем шаге рассматриваемого двухкритериального алгоритма предлагается процедура планирования, которая состоит из двух этапов.

На первом этапе для каждого информационного выхода системы определяются подсистемы, включающие в себя один выходной процессор системы и процессоры, реализующие выполнение заданий, соответствующих этому выходу.

При этом заданием называется совокупность n задач, связанных отношением предшествования. К подсистемам такого вида могут быть применены алгоритмы *flow shop*-планирования.

Рассмотрим распределенную вычислительную систему $S(P, \tau)$, состоящую из множества процессоров $P = \{P_i | i = \overline{1, n}\}$, обладающих одинаковой производительностью. Процессоры выполняют определенные задачи и по готовности передают результат следующему процессору по каналу передачи данных. Предположим, что планированию в системе подлежат m независимых равноприоритетных заданий $\tau = \{\tau_j | j = \overline{1, m}\}$, поступающих с периодом T . Задачи обозначим $\tau_{j,i}$, а их длительность — $\tau_{j,i}$, $i = \overline{1, n}$.

Поскольку назначение выполнено на предыдущем этапе алгоритма, считаем, что имеется m изоморфизмов $\phi_j: G_j(E_j, T_j) \rightarrow H(Q, P)$, $j = \overline{1, m}$, где $G_j(E_j, T_j)$ — граф межзадачных связей j -го задания, E_j — множество ребер, T_j — множество вершин (задач), $H(Q, P)$ — граф межпроцессорных связей, Q — множество ребер.

В рамках предложенного подхода частный план составляется для каждой из подсистем. Для составления частных планов предлагается использовать алгоритмы, в основе которых лежит РКС-формализм [9]. Суть его заключается в выделении четырех разрешимых классов систем (РКС), для трех из которых предлагаются оптимальные алгоритмы планирования, а для четвертого — субоптимальный. Эти алгоритмы обладают полиномиальной вычислительной сложностью, что является их преимуществом по сравнению с другими алгоритмами, например алгоритмом перебора.

В основе определения разрешимых классов лежит понятие отношения доминирования ($>$): процессор P_q доминирует над процессором P_r ($P_q > P_r$), если минимальное время выполнения задач на процессоре P_q больше максимального времени выполнения задач на процессоре P_r — $P_q - \min_j e_{j,q} \geq \max_j e_{j,r}, j = \overline{1, m}$) [9].

Для любого задания, выполняемого в системе, относящейся к разрешимому классу, критический путь единствен и проходит по одним и тем же процессорам $P_{[1]}, P_{[2]}, \dots, P_{[m^*]}$, где $[1], [2], \dots, [m^*]$ — номера процессоров критического пути. Разрешимые классы систем отличаются характерными упорядоченностями процессоров критического пути системы относительно

доминирования: убывающая последовательность — для первого класса, возрастающая — для второго, соединенные возрастающая и убывающая последовательности — для третьего, соединенные убывающая и возрастающая последовательности — для четвертого.

Ниже представлены алгоритмы *flow shop*-планирования для разрешимых классов, позволяющие построить план π выполнения заданий при использовании в качестве критерия J минимума среднего по заданиям времени пребывания задания в системе $\bar{F}(\pi)$ [8]:

$$J = \min_{\pi} \bar{F}(\pi).$$

Алгоритм 1. Составление частного плана системы, относящейся к первому разрешимому классу.

Шаг 1. Определить для рассматриваемой системы критический путь.

Шаг 2. Упорядочить задания в плане π в соответствии со следующим правилом для задач критического пути:

$$e_{1,[1]}^* \leq e_{2,[1]}^* \leq \dots \leq e_{n,[1]}^*,$$

где $e_{i,[1]}^*$ — длительность первых задач критического пути.

Алгоритм 2. Составление частного плана системы, относящейся ко второму разрешимому классу.

Шаг 1. Определить для рассматриваемой системы критический путь.

Шаг 2. Упорядочить задания в плане π в соответствии со следующим правилом для задач критического пути:

$$e_{1,[m^*]}^* \leq e_{2,[m^*]}^* \leq \dots \leq e_{n,[m^*]}^*,$$

Шаг 3. Проверить первое задание плана π на соответствие условию:

$$j^* = \arg \min_j \sum_{i=1}^{m^*-1} e_{j,[i]}^*.$$

Алгоритм 3. Составление частного плана системы, относящейся к третьему разрешимому классу.

Шаг 1. Определить для рассматриваемой системы критический путь.

Шаг 2. Упорядочить задания в плане π в соответствии со следующим правилом для задач критического пути:

$$e_{1,[h^*]}^* \leq e_{2,[h^*]}^* \leq \dots \leq e_{n,[h^*]}^*.$$

Шаг 3. Проверить первое задание плана π на соответствие условию:

$$j^* = \arg \min_j \sum_{i=1}^{h^*-1} e_{j,[i]}^*.$$

Алгоритм 4. Составление частного плана системы, относящейся к четвертому разрешимому классу.

Шаг 1. Определить для рассматриваемой системы критический путь.

Шаг 2. Упорядочить задания в плане π в соответствии со следующим правилом для задач критического пути:

$$(e_{1,[1]}^* + e_{1,[m^*]}^*) \leq (e_{2,[m^*]}^* + e_{2,[m^*]}^*) \leq \dots \leq (e_{n,[1]}^* + e_{n,[m^*]}^*).$$

Шаг 3. Проверить первое задание плана π на соответствие условию:

$$j^* = \arg \min_j \sum_{i=h^*+1}^{m^*-1} e_{j,[i]}^*.$$

Если в конкретном случае условия в алгоритмах 2–4 противоречат друг другу, то лучший из вариантов может быть определен перебором.

Вычислительную сложность приведенных алгоритмов можно оценить как $O(n \log n)$.

Интеграция частных планов системы flow shop в результирующий план. Частные планы, полученные для каждой подсистемы, необходимо интегрировать в общий план для всей системы. Для решения этой задачи предлагается использовать эвристический алгоритм, основанный на НЕН-алгоритме [5]. Суть алгоритма заключается в последовательном размещении в плане предварительно упорядоченных заданий.

Алгоритм 5. Планирование вычислений в системе со многими выходами, по критерию минимума суммарного времени пребывания задания в системе.

Шаг 1. Упорядочить сформированные подсистемы по количеству исполняемых ими заданий: от подсистемы, выполняющей наибольшее число заданий, к подсистеме, выполняющей наименьшее число.

Шаг 2. Определить текущий план для системы как план для подсистемы, выполняющей наибольшее число заданий.

Шаг 3. Построить для системы интегрированный план, последовательно дополняя текущий план заданиями из других частных планов, выбирая эти планы в соответствии с упорядоченностью, полученной на шаге 1. При включении в интегрированный план заданий из очередного частного плана использовать НЕН-алгоритм, сохраняя принятую в частном плане упорядоченность.

Двухкритериальный алгоритм, обеспечивающий энергоэффективное планирование в системе на основе рассмотренных выше алгоритмов, имеет следующий вид.

Алгоритм 6. Энергоэффективное планирование информационных процессов в вычислительной системе.

Шаг 1. Определить энергоэффективную архитектуру системы в соответствии с алгоритмом [12].

Шаг 2. Если полученная система имеет несколько информационных выходов — выделить подсистемы, соответствующие каждому информационному выходу, и осуществить планирование выполняемых в них заданий с использованием алгоритмов 1–4, после чего составить общий план с помощью алгоритма 5. Если архитектура полученной системы содержит один информационный выход — осуществить планирование выполняемых в ней заданий с использованием алгоритмов 1–4.

Апробация алгоритма энергоэффективного планирования. Для успешного выполнения задач, стоящих перед АНПА, необходима высокоточная навигация [15]. Обычно навигация обеспечивается бортовыми навигационными системами, такими как инерциальная навигационная, чаще — беспилотная (БИНС), и измеритель скорости движения (лаг). Тем не менее, в силу накапливающейся погрешности счисления требуется периодическая коррекция координат АНПА [16].

Рассмотрим АНПА, в котором для получения высокой точности определения местоположения применяется навигационный комплекс, включающий, помимо БИНС и относительного лага, эхолот, гравиметр, блок навигации по геофизическим полям (ГФП), а также гидроакустическую навигационную систему (ГАНС). Информация, получаемая от перечисленных систем, преобразуется при помощи средств цифровой обработки (СЦОИ).

Схема рассматриваемого макета навигационного комплекса АНПА приведена на рис. 1.

Рассмотрим пример применения алгоритма энергоэффективного планирования при выполнении вычислений в навигационном комплексе, а именно при выполнении задач, приведенных на схеме вычислительной системы навигационного комплекса на рис. 2.*

Отобразим на схеме вычислительной системы выполняемые в ней четыре задания, по числу входных процессоров (τ_1 – τ_4). Схема выполнения заданий на процессорах вычислительной системы навигационного комплекса АНПА приведена на рис. 3.

* Помимо отображенных на схеме существуют и другие задачи, которые могут быть добавлены в план при помощи иных алгоритмов.



Рис. 1

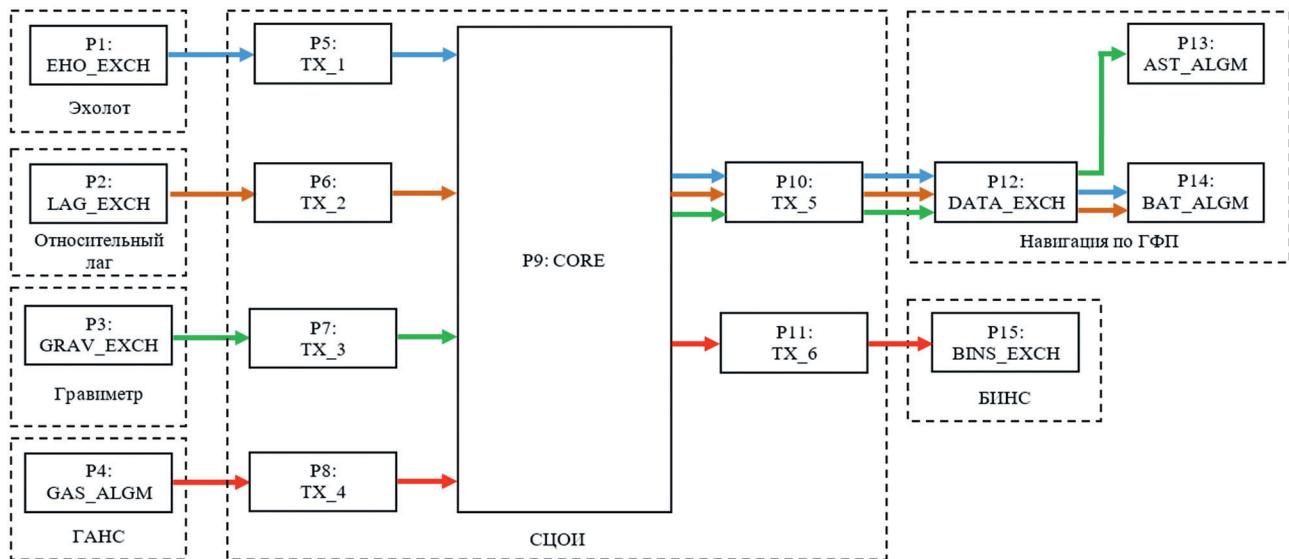


Рис. 2

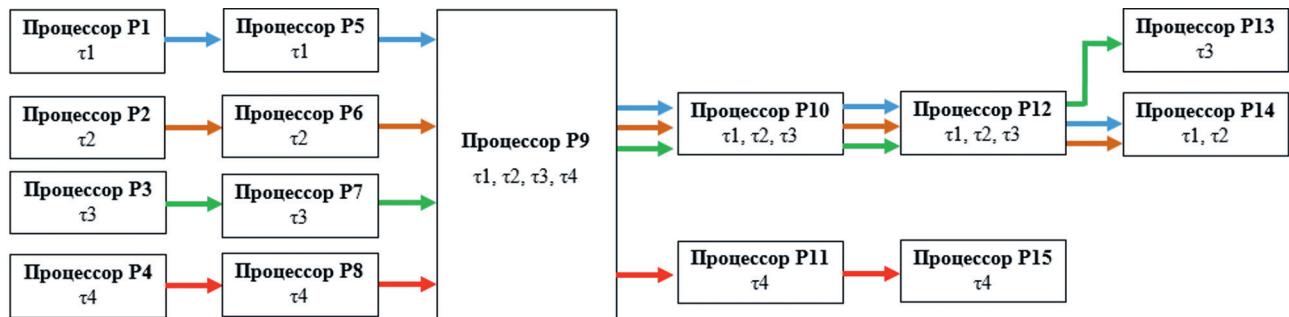


Рис. 3

Длительность задач и потребляемая на каждом процессоре мощность приведены в табл. 1.

Определим энергоэффективную архитектуру вычислительной системы навигационного комплекса АНПА, предположив, что в нее можно добавить два процессора. В соответствии с шагами алгоритма [12]:

1) определим исходные данные: $M = 2$, $A = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)$, $\bar{W} = (4 \ 6 \ 4 \ 10 \ 2 \ 2 \ 2 \ 2 \ 26 \ 6 \ 2 \ 12 \ 4 \ 8 \ 6)$;

2) очевидно, самым загруженным является процессор P9 и потребляемая им мощность является максимальной, поэтому разделим процессор P9 на два, распределив задачи следующим

образом: $P_{9,1} = \tau_{1,9}$, $\tau_{2,9}$ и $\tau_{3,9}$; $P_{9,2} = \tau_{4,9}$. Коэффициент снижения тактовой частоты и напряжения питания составит $q_9 = \frac{\tau_{\Sigma,9}}{\tau_{1,9} + \tau_{2,9} + \tau_{3,9}} = 1,44$. Новое значение потребляемой мощности на разделенном процессоре P_9 составляет $\bar{P}_9' = \frac{\bar{P}_9}{q_9^3} = 8,7$. Количество оставшихся запасных процессоров $M = 1 \neq 0$;

3) теперь наибольшим энергопотреблением обладает процессор P_{12} . Разделим процессор P_{12} : $P_{12,1} = \tau_{1,12}$ и $\tau_{3,12}$, $P_{12,2} = \tau_{2,12}$. Коэффициент снижения тактовой частоты и напряжения питания $q_{12} = \frac{\tau_{\Sigma,12}}{\tau_{1,12} + \tau_{3,12}} = 2$. Новое значение потребляемой мощности на разделенном процессоре P_{12} составляет $\bar{W}_{12}' = \frac{\bar{W}_{12}}{q_{12}^3} = 1,5$. Количество оставшихся запасных процессоров $M = 0$.

Таблица 1

Параметр	Длительность, усл. ед.														
	$e_{j,1}$	$e_{j,2}$	$e_{j,3}$	$e_{j,4}$	$e_{j,5}$	$e_{j,6}$	$e_{j,7}$	$e_{j,8}$	$e_{j,9}$	$e_{j,10}$	$e_{j,11}$	$e_{j,12}$	$e_{j,13}$	$e_{j,14}$	$e_{j,15}$
τ_1	2	0	0	0	1	0	0	0	3	1	0	2	0	2	0
τ_2	0	3	0	0	0	1	0	0	4	1	0	3	0	2	0
τ_3	0	0	2	0	0	0	1	0	2	1	0	1	2	0	0
τ_4	0	0	0	5	0	0	0	1	4	0	1	0	0	0	3
\bar{e} (медианы)	0	0	0	0	0	0	0	0	3,5	1	0	1,5	0	1	0
e_{Σ}	2	3	2	5	1	1	1	1	13	3	1	6	2	4	3
Средняя потребляемая на процессоре мощность, усл. ед.															
\bar{W}	4	6	4	10	2	2	2	2	26	6	2	12	4	8	6

Теперь необходимо определить эффективность алгоритма определения энергоэффективной архитектуры [12] путем сравнения полученного значения потребляемой системой мощности с исходным.

Потребляемая исходной системой мощность вычисляется следующим образом:

$$W_0 = \sum_j \bar{W}_j = 96.$$

Для преобразованной системы выражение будет иметь вид:

$$W' = \sum_j \frac{\bar{W}_{\Sigma,j}}{k_j^3} k = \sum_j \bar{W}_j' k = 78,4,$$

где $k = \frac{n + n_d}{n_d}$ — коэффициент, отвечающий за увеличение числа процессоров.

Таким образом, разность полученных значений составит:

$$\Delta = \frac{P_0 + P'}{P_0} = 18,3 \text{ \%}.$$

Схема полученной системы приведена на рис. 4, а длительности задач на процессорах и потребляемые мощности приведены в табл. 2.

Перейдем к следующему шагу алгоритма 6. Полученная система имеет три информационных выхода. Следовательно, необходимо выделить в ней подсистемы и построить частные планы для каждой из них.

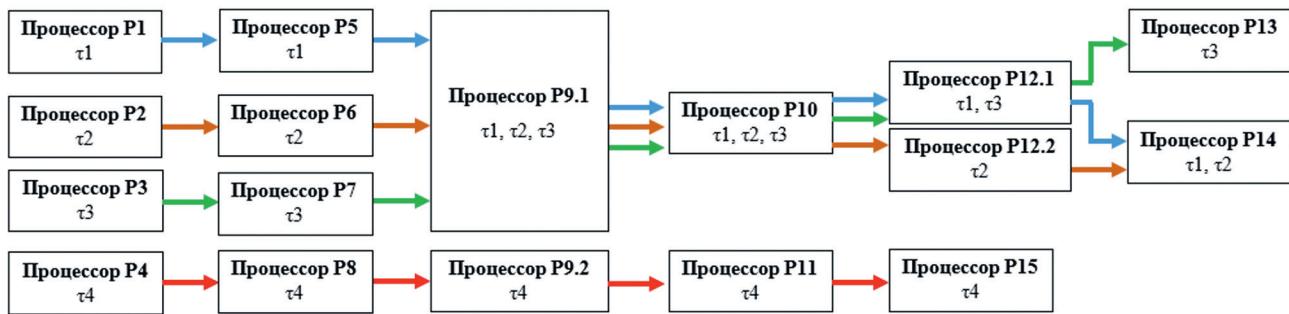


Рис. 4

Таблица 2

Параметр	Длительность, усл. ед.																	
	$e_{j,1}$	$e_{j,2}$	$e_{j,3}$	$e_{j,4}$	$e_{j,5}$	$e_{j,6}$	$e_{j,7}$	$e_{j,8}$	$e_{j,9.1}$	$e_{j,9.2}$	$e_{j,10}$	$e_{j,11}$	$e_{j,12.1}$	$e_{j,12.2}$	$e_{j,13}$	$e_{j,14}$	$e_{j,15}$	
τ_1	2	0	0	0	1	0	0	0	3	0	1	0	2	0	0	2	0	
τ_2	0	3	0	0	0	1	0	0	4	0	1	0	0	3	0	2	0	
τ_3	0	0	2	0	0	0	1	0	2	0	1	0	1	0	2	0	0	
τ_4	0	0	0	5	0	0	0	1	0	4	0	1	0	0	0	0	3	
\bar{e} (медианы)	0	0	0	0	0	0	0	0	2,5	0	1	0	0,5	0	0	1	0	
e_{Σ}	2	3	2	5	1	1	1	1	9	4	3	1	3	3	2	4	3	
Средняя потребляемая на процессоре мощность, усл. ед.																		
\bar{W}	4	6	4	10	2	2	2	2	8,7	6	2	1,5	4	8	6			

В полученной системе можно выделить три подсистемы для каждого информационного выхода:

— в подсистеме S1, включающей следующие процессоры: P3 – P7 – P9.1 – P10 – P12.1 – P13, выполняется задание T3;

— в подсистеме S2, включающей следующие процессоры: P1, P2 – P5, P6 – P9.1 – P10 – P12.1, P12.2 – P14, выполняются задания T1, T2;

— в подсистеме S3, включающей следующие процессоры: P4 – P8 – P9.2 – P11 – P15, выполняется задание T4.

Поскольку в подсистеме S2 выполняется два задания, необходимо их предварительно упорядочить. Для остальных подсистем построение частного плана не требуется. В подсистеме S2 существует два вычислительных пути, произведем расчет длительности для определения того, какой из них является псевдокритическим.

Первый путь включает процессоры P1, P5, P9.1, P10, P12.1, P14 и его длительность составляет:

$$\bar{e}(p_{1,5,9.1,10,12.1,14}) = 0 + 0 + 2,5 + 1 + 0,5 + 1 = 5.$$

Второй путь включает в себя процессоры P2, P6, P9.1, P10, P12.2, P14 и его длительность составляет:

$$\bar{e}(p_{2,6,9.1,10,12.2,14}) = 0 + 0 + 2,5 + 1 + 0 + 1 = 4,5.$$

Таким образом, псевдокритическим является более длительный первый путь. В соответствии с длительностью выполняемых на процессорах критического пути задач с использованием геометрического правила [9] определено, что подсистема S2 относится к третьему классу, следовательно, для упорядочивания заданий в ней необходимо использовать алгоритм 4.

Процессором стыковки является Р9.1. Длительность задачи из задания τ_1 на процессоре стыковки составляет 3, а для задания τ_2 — 4. При этом общее время пребывания задания τ_1 в системе составляет 11, а τ_2 — 14. Таким образом, для подсистемы S2 частный план имеет следующий вид — (τ_1, τ_2) . Задания τ_3 и τ_4 в плане размещены последовательно с использованием алгоритма NEH: при сохранении относительного размещения в плане заданий τ_1 и τ_2 сначала размещено задание τ_3 на разных позициях и произведено сравнение среднего времени пребывания в системе заданий в соответствии с каждым планом, получен план с минимальным средним временем выполнения заданий (τ_3, τ_1, τ_2) . Задание τ_4 выполняется параллельно с остальными, поэтому его размещение в плане не влияет на среднее время пребывания заданий в системе. Таким образом, среднее время выполнения заданий в системе с энергоэффективной архитектурой составило 13,5. При этом значение среднего времени пребывания заданий в исходной системе с оптимальным планом, определенным методом перебора, составило 15.

В результате апробации предложенный алгоритм позволил снизить энергопотребление системы на 18,3 %, а также снизить и среднее время пребывания заданий на 10 % при снижении вычислительной сложности используемого алгоритма планирования заданий.

Заключение. В работе предложен позволяющий снизить энергопотребление системы и сократить среднее время пребывания заданий в системе двухкритериальный алгоритм энергоэффективного планирования заданий, отличающийся от известных возможностью планирования в системах со многими информационными выходами. Апробация предложенного алгоритма на примере вычислительной системы навигационного комплекса показала его эффективность. Использование алгоритма позволило снизить энергопотребление системы на 18,3 % и среднее время пребывания заданий на 10 %, по сравнению с оптимальными значениями энергопотребления и времени в исходной системе.

СПИСОК ЛИТЕРАТУРЫ

1. Инзарцев А. В., Киселев Л. В., Костенко В. В., Матвиенко Ю. В., Павин А. М., Щербатюк А. Ф. Подводные робототехнические комплексы: системы, технологии, применение. Владивосток: Ин-т проблем морских технологий ДВО РАН, 2018. 368 с.
2. Ramirez I. S., Bernalte Sánchez P. J., Papaelias M., Márquez F. P. G. Autonomous underwater vehicles and field of view in underwater operations // Journal of Marine Science and Engineering. 2021. Vol. 9, N 3. P. 277.
3. Yang Y., Xiao Y., Li T. A survey of autonomous underwater vehicle formation: Performance, formation control, and communication capability // IEEE Communications Surveys & Tutorials. 2021. Vol. 23, N 2. P. 815–841.
4. Маевский А. М., Гайкович Б. А. Разработка легкого интервенционного автономного необитаемого подводного аппарата в целях использования в подводных резидентных системах // Матер. XIV Всерос. науч.-практ. конф. и X молодежной школы-семинара „Управление и обработка информации в технических системах“. Ростов-на-Дону, 2019. С. 83.
5. Nawaz M., Enscore E. E., jr., Ham I. A Heuristic Algorithm for the m-Machine, n-Job Flow-shop Sequencing Problem // Omega — Intern. J. of Management Science. 1983. N 11. P. 91–95.
6. Kshemkalyani A. D., Singhal M. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011. 731 p.
7. Малашенко Ю. Е., Назарова И. А. Управление ресурсоемкими разнородными вычислительными заданиями с директивными сроками окончания // Изв. РАН. ТиСУ. 2012. № 5. С. 15–22.
8. Лазарев А. А., Гафаров Е. Р. Теория расписаний. Задачи и алгоритмы. М.: МГУ, 2011. 222 с.
9. Колесов Н. В., Толмачева М. В., Юхта П. В. Системы реального времени. Планирование, анализ, диагностирование. СПб: Электроприбор, 2014. 185 с.
10. Brucker P. Scheduling Algorithms. Springer Berlin, Heidelberg, 2007. 371 p.
11. Колесов Н. В., Грузликов А. М., Скородумов Ю. М., Толмачева М. В. Смешанное планирование заданий в распределенных системах реального времени // Вестник компьютерных и информационных технологий. 2016. № 5(143). С. 34–40.
12. Грузликов А. М., Колесов Н. В., Костыгов Д. В., Ошуев В. В. Энергоэффективное планирование в распределенных вычислительных системах реального времени // Изв. Российской академии наук. Теория и системы управления, 2019. №. 3. С. 66–76.

13. Panda P. R., Shrivastava A., Silpa B. V. N., Gummidipudi K. Power-efficient System Design. NY: Springer, 2010. 260 p.
14. Колесов Н. В., Литуненко Е. Г., Скородумов Ю. М., Толмачева М. В. Планирование заданий в распределенной вычислительной системе на кристалле с минимизацией потребляемой мощности // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 5. С. 1001–1008.
15. Спасский Б. А. Автономная навигация необитаемых подводных аппаратов // Робототехника и техническая кибернетика. 2014. № 4. С. 13–20.
16. Машошин А. И., Пашкевич И. В. Применение подводных пассивных ориентиров для навигации автономных необитаемых подводных аппаратов // Гироскопия и навигации. 2020. Т. 28, № 3. С. 110.

СВЕДЕНИЯ ОБ АВТОРЕ

Елизавета Геннадьевна Литуненко — аспирант; ЦНИИ „Электроприбор“; E-mail: lisa.litunenko@gmail.com

Поступила в редакцию 10.07.24; одобрена после рецензирования 09.09.24; принята к публикации 21.11.24.

REFERENCES

1. Inzartsev A.V., Kiselev L.V., Kostenko V.V., Matvienko Yu.V., Pavin A.M., Shcherbatuk A.F. *Podvodnyye robototekhnicheskiye kompleksy: sistemy, tekhnologii, primeneniye* (Underwater Robotic Systems: Systems, Technologies, Applications), Vladivostok, 2018, 368 p. (in Russ.)
2. Ramírez I.S., Bernalte Sánchez P.J., Papaelias M., Márquez F.P.G. *Journal of Marine Science and Engineering*, 2021, no. 3(9), pp. 277.
3. Yang Y., Xiao Y., Li T. *IEEE Communications Surveys & Tutorials*, 2021, no. 2(23), pp. 815–841.
4. Maevsky A.M., Gaikovich B.A. *Upravleniye i obrabotka informatsii v tekhnicheskikh sistemakh* (Management and Processing of Information in Technical Systems), Proceedings of the XIV All-Russian Scientific and Practical Conference and X Youth School-Seminar, Rostov-on-Don, 2019, pp. 83. (in Russ.)
5. Nawaz M., Enscore E.E., jr., Ham I. *Omega – International Journal of Management Science*, 1983, no. 11, pp. 91–95.
6. Kshemkalyani A.D., Singhal M. *Distributed computing: principles, algorithms, and systems*, Cambridge University Press, 2011, 731 p.
7. Malashenko Yu.E., Nazarova I.A. *Journal of Computer and Systems Sciences International*, 2012, no. 5, pp. 628–635.
8. Lazarev A.A., Gafarov E.R. *Teoriya raspisaniy. Zadachi i algoritmy* (Scheduling Theory. Problems and Algorithms), Moscow, 2011, 222 p. (in Russ.)
9. Kolesov N.V., Tolmacheva M.V., Yukhta P.V. *Sistemy real'nogo vremeni. Planirovaniye, analiz, diagnostirovaniye* (Real-time Systems. Planning, Analysis, Diagnostics), St. Petersburg, 2014, 185 p. (in Russ.)
10. Brucker P. *Scheduling Algorithms*, Springer Berlin, Heidelberg, 2007, 371 p.
11. Kolesov N.V., Gruzlikov A.M., Skorodumov Yu.M., Tolmacheva M.V. *Vestnik komp'iuternykh i informatsionnykh tekhnologii*, 2016, no. 5(143), pp. 34–40.
12. Gruzlikov A.M., Kolesov N.V., Kostygov D.V., Oshuev V.V. *Journal of Computer and Systems Sciences International*, 2019, no. 3, pp. 393–403.
13. Panda P.R., Shrivastava A., Silpa B.V.N., Gummidipudi K. *Power-efficient System Design*, NY, Springer, 2010, 260 p.
14. Kolesov N.V., Litunenko E.G., Skorodumov Yu.M., Tolmacheva M.V. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, no. 5(23), pp. 1001–1008. (in Russ.)
15. Spassky B.A. *Robotics and Technical Cybernetics*, 2014, no. 4, pp. 13–20. (in Russ.)
16. Mashoshin A.I., Pashkevich I.V. *Gyroscopy and Navigation*, 2020, no. 3(28), pp. 110. (in Russ.)

DATA ON AUTHOR

Elizaveta G. Litunenko

— PhD Student; Concern CSRI Elektropribor, JSC;
E-mail: lisa.litunenko@gmail.com

Received 10.07.24; approved after reviewing 09.09.24; accepted for publication 21.11.24.