УДК 621.398

С.В. Минухин, канд. техн. наук
Харьковский национальный экономический университет
(Украина, 61001, Харьков, пр-т Ленина, 9-а,
тел.: 057 7021831, e-mail: ms_vl@mail.ru),
Д.С. Ленько
(Украина, Харьков, e-mail: dmytrolenko@gmail.com)

Метод минимизации суммарного запаздывания работ на одиночном устройстве на основе рангового подхода и правил доминирования

Рассмотрен метод минимизации суммарного запаздывания работ на одиночном устройстве на основе определения кратчайшего гамильтонового пути в произвольном полносвязном графе с использованием рангового подхода и правил доминирования. Предложены метрики оценивания улучшения результатов работы алгоритма при использовании правил доминирования — относительного уменьшения суммарного запаздывания и числа получаемых локально-оптимальных решений. Приведены результаты вычислительных экспериментов для взвешенного и невзвешенного случаев запаздывания работ с произвольными директивными сроками. Найдены условия, при которых используемый алгоритм позволяет улучшить расписания работ и определить наиболее эффективные правила доминирования.

Розглянуто метод мінімізації сумарного запізнювання робіт на одиночному пристрої на основі визначення найкоротшого гамільтонового шляху в довільному повнозв'язному графі з використанням рангового підходу і правил домінування. Запропоновано метрики оцінювання поліпшення результатів роботи алгоритму при використанні правил домінування — відносного зменшення сумарного запізнювання і кількості одержуваних локально-оптимальних рішень. Наведено результати обчислювальних експериментів для зваженого і незваженого запізнювання робіт з довільними директивними термінами. Визначено умови, при яких досліджуваний алгоритм дозволяє покращити розклад робіт, і знайдено найбільш ефективні правила домінування.

Ключевые слова: ранговый подход, гамильтонов путь, расписание, суммарное запаздывание, временная сложность, директивный срок, одиночное устройство, длительность, правило доминирования.

Для современных информационно-коммуникационных систем характерно наличие большого числа различных типов ресурсов — информационных, вычислительных и других, управление которыми требует повышения эффективности и совершенствования систем управления распреде-

ленной обработкой. Одной из важнейших задач является разработка эффективных алгоритмов построения локальных расписаний выполнения работ в соответствии с выбранными критериями на вычислительных устройствах, таких, например, как узлы Грид-систем — вычислительные кластеры, кластеры рабочих станций, некластеризованные ресурсы. В качестве таких критериев могут быть использованы ограничения на время выполнения потоков заданий или бюджетные расходы, в основу которых положена модель работы с директивным сроком выполнения, требующая новых подходов к решению задач планирования.

В связи с этим получили развитие методы построения расписаний выполнения работ, широко используемые в промышленном планировании. В [1] предложена модель работы веб-сервера, использующего одну очередь работ. Эта модель позволяет дифференцировать QoS (Quality of Service) для различных классов запросов к веб-сервисам по таким приоритетам, как взвешенное наименьшее время выполнения (WSPT), наименьший директивный срок (EDD) и др. Эти правила использованы при оценке QoS для стратегии FIFO (FCFS) в модели Best Effort, в которой веб-запрос описывается приоритетом, временем поступления на обработку, требуемой для его реализации памятью, и директивным сроком выполнения. Базовая модель Best Effort включает множество запросов, очереди для их сортировки и ожидания запросов в очереди до времени их обслуживания на сервере.

В работах [2—4] рассмотрены подходы к планированию непрерывного потока пакетов заданий с использованием правил приоритета на машинах (процессорных элементах) распределенной Грид-системы. Новизна решения, предложенного в [4—6], заключается в использовании правила «наиболее ранний зазор EG (Earliest Gap) — первый с наиболее ранним директивным сроком EDF (Earliest Deadline First)» и запрет на поиск, основанный на идее заполнения «зазоров» в текущем расписании работ. Правило EG— EDF позволяет строить расписание для всех работ последовательно, применяя технику, с помощью которой заполняются ранее имеющиеся зазоры — EG — в расписании из множества вновь прибывших работ. В случае, если зазор для поступающей работы отсутствует, то используется второе правило — EDF — для включения новой работы в существующее расписание.

Для распределенной вычислительной системы (Грид-системы, кластера) используется метод оптимизации локальных расписаний [7, 8], предусматривающий применение режима пакетного планирования, который состоит в следующем. Из входного потока работ формируется пул, размер

которого определяется на основании характеристик распределенной среды и работ. Затем планировщик планирует (назначает) определенные ресурсы для их выполнения в соответствии с моделью метода о наименьшем покрытии и отправляет задания на полученные таким образом ресурсы в виде множества пакетов, упорядоченных и организованных в виде очередей, каждая из которых должна выполняться на ресурсе в условиях ограничений (на бюджет или использование директивного срока). Если ограничением является время выполнения или директивный срок работы, то можно использовать методы построения расписания выполнения заданий на одиночном устройстве (приборе), учитывая специфику работы описанного механизма системы планирования.

Эта задача, с формальной точки зрения, может быть сведена, например, к классической задаче минимизации суммарного (среднего) взвешенного и невзвешенного запаздывания работ на одном устройстве (приборе). Ее решение вызывает практический интерес относительно оптимизации (повышения эффективности) функционирования локальных менеджеров ресурсов распределенных систем с целью уменьшения, а возможно, и устранения запаздывания в случаях, когда в модели, описывающей работу, используется директивный срок. Это позволит снизить риски, связанные со штрафными санкциями, приводящими к экономическим убыткам заказчиков и пользователей вычислительных ресурсов.

В работах [9—18] приведены методы решения задачи минимизации суммарного запаздывания, а в работах [19—20] предложен и исследован эффективный точный полиномиальной сложности алгоритм решения задачи минимизации суммарного запаздывания одним прибором. В [21] представлен метод минимизации суммарного времени запаздывания (оптимизации по направлению), основанный на построении кратчайшего гамильтонова пути в полносвязном произвольном графе и рангового подхода с временной сложностью $\boldsymbol{O}\ (n^3)$, где n — число работ (вершин в графе).

В работах [22, 23] выдвинута гипотеза о том, что в случаях, когда на ранге существует несколько альтернативных путей, имеющих одинаковую длину, нужно выбирать тот путь, в котором последовательность работ (вершин), предшествующих данной вершине, удовлетворяет правилу доминирования.

Исследуем метод и алгоритм минимизации суммарного времени запаздывания работ на одиночном вычислительном устройстве на основе построения кратчайшего гамильтонова пути в произвольном графе и рангового подхода с использованием правил доминирования и оценки его эффективности для работ с произвольными директивными сроками.

Постановка задачи. Предположим, что на одиночное устройство (вычислительный ресурс) поступает множество независимых работ

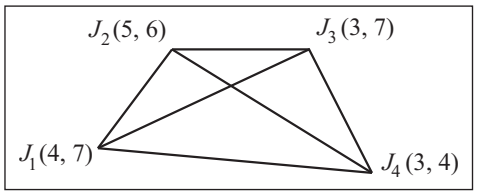

Рис. 1. Полносвязный граф из четырех вершин (работ)

$J = \{J_1, J_2, ..., J_n\}$, каждая из которых будет непрерывно (без прерываний) выполняться на нем. При этом известны длительность выполнения каждого задания L_j и директивный срок его выполнения d_j. Необходимо определить такой порядок (последовательность) выполнения всех заданий, одновременно поступающих на устройство, который минимизирует суммарное время TT_S запаздывания всех заданий входной очереди,

$$TT_S = \sum_{\{j \in S\}} \max(0, C_j - d_j),$$

где C_j — время завершения работы j. Метод решения этой задачи, обозначаемой $1||\sum T_i$, предложен в [21].

Задачу минимизации $(1||\sum w_i T_i)$ суммарного взвешенного времени запаздывания на одиночном устройстве,

$$TWT_S = \sum_{\{j \in S\}} \max(0, (C_j - d_j)\, w_j),$$

сформулируем так. Множество работ, проиндексированных от 1 до n, должно быть выполнено без прерываний на одиночном устройстве, обрабатывающем только одну работу в определенный момент времени. Все работы поступают на устройство в момент времени $t = 0$ (или одновременно). Работа характеризуется временем ее обработки L_i (или процессорным временем p_i), директивным сроком d_i и весом w_i. Для удобства все работы упорядочены согласно правилу EDD, поэтому $d_i < d_j$. Если $d_i = d_j$, то $p_i < p_j$; если $p_i = p_j$, то $w_i > w_j$ для всех i, j $(i < j)$. Если работа завершается позже своего директивного срока d_i, результатом является взвешенный штраф за запаздывание.

Для обоснования целесообразности использования правил доминирования рассмотрим работу алгоритма на основе произвольного полносвязного графа с четырьмя вершинами, которые определяют работы тестовой последовательности, суть метода и использование правил доминирования.

Пусть для множества работ J_1 (4, 7), J_2 (5, 6), J_3 (3, 7), J_4 (3, 4) необходимо построить оптимальное расписание. Каждая работа J имеет два параметра: длительность L и директивный срок d. Следовательно, ее можно представить в виде J (L, d). Граф, отображающий данное множество работ, представлен на рис. 1.

Для пояснения работы базового алгоритма [21] используем матрицу, в которой столбцам соответствуют ранги, а строкам — работы. Для этих

Таблица 1. **Пример работы алгоритма согласно правилу EDD**

Вершина	Ранг 1	Ранг 2	Ранг 3	Ранг 4
$J_1(4,7)$	$T(1)_1=(0+4)-7=-3$	$T(2)_{2,1}=(5+4)-7=2$	$T(3)_{4,2,1}=(8+4)-7=5$	$TT_{3,4,2,1}=0+2+5+8=15$
		$T(2)_{3,1}=(3+4)-7=0$	$T(3)_{4,3,1}=(6+4)-7=3$	
		$\boldsymbol{T(2)_{4,1}=(3+4)-7=0}$	$\boldsymbol{T(3)_{3,4,1}=(6+4)-7=3}$	
$J_2(5,6)$	$T(1)_2=(0+5)-6=-1$	$T(2)_{1,2}=(4+5)-6=3$	$T(3)_{4,1,2}=(7+5)-6=6$	$\boldsymbol{TT_{4,3,1,2}=0+0+3+9=12}$
		$T(2)_{3,2}=(3+5)-6=2$	$T(3)_{4,3,2}=(6+5)-6=5$	$TT_{4,1,3,2}=0+0+3+9=12$
		$\boldsymbol{T(2)_{4,2}=(3+5)-6=2}$	$\boldsymbol{T(3)_{3,4,2}=(6+5)-6=5}$	
$J_3(3,7)$	$T(1)_3=(0+3)-7=-4$	$T(2)_{1,3}=(4+3)-7=0$	$\boldsymbol{T(3)_{4,1,3}=(7+3)-7=3}$	
		$T(2)_{2,3}=(5+3)-7=1$	$T(3)_{4,2,3}=(8+3)-7=4$	
		$\boldsymbol{T(2)_{4,3}=(3+3)-7=-1}$		
$J_4(3,4)$	$T(1)_4=(0+3)-10=-7$	$T(2)_{1,4}=(4+3)-4=3$		
		$T(2)_{2,4}=(5+3)-4=4$		
		$\boldsymbol{T(2)_{3,4}=(3+3)-4=2}$		

Примечание. Полужирным шрифтом выделены оптимальные пути, определяемые алгоритмом на каждом ранге.

работ необходимо построить график выполнения (табл. 1). Величину текущего запаздывания работ на каждом ранге обозначим $T(J)_{j_1 j_2 \ldots j_n}$, где J — число рангов (столбцов), $j_1, j_2, \ldots, j_n$ — вершины (работы), входящие в текущий путь к какой-либо вершине (работе) на текущем ранге (столбце) матрицы. Таким образом, суммарное запаздывание всех работ TT будет определяться на последнем ранге.

В начальном состоянии работы не упорядочены, запаздывание каждой из них равно нулю. В первой строке матрицы находятся все пути от работ J_2, J_3, J_4 к работе J_1, во второй — пути от работ J_1, J_3, J_4 к работе J_2, в третьей — пути от работ J_1, J_2, J_4 к работе J_3, в четвертой — пути от работ J_1, J_2, J_3 к работе J_4. Далее, в соответствии с алгоритмом, необходимо для каждой строки выбрать минимальный путь:

для первой строки — $T(2)_{3,1} = 3 + (4–7) = 0$ и $T(2)_{4,1} = 3 + (4–7) = 0$;
для второй строки — $T(2)_{3,2} = 3 + (5–6) = 2$ и $T(2)_{4,2} = 3 + (5–6) = 2$;
для третьей строки — $T(2)_{4,3} = 3 + (3–7) = –1$;
для четвертой строки — $T(2)_{3,4} = 3 + (3–4) = 2$.

После этого на ранге 3 выбираем:

в первой строке — $T(3)_{4,3,1} = 6 + (4–7) = 3$ и $T(3)_{3,4,1} = 6 + (4–7) = 3$;
во второй строке — $T(3)_{4,3,2} = 6 + (5–6) = 5$ и $T(3)_{3,4,2} = 6 + (5–6) = 5$;
в третьей строке — $T(3)_{4,1,3} = 7 + (3–7) = 3$;
в четвертой строке — $T(3)_{3,1,4} = 7 + (3–4) = 6$.

Далее строим пути из вершин ранга 3 к вершинам ранга 4:

для первой строки — $TT_{4,3,2,1} = 0 + 0 + 5 + 8 = 13$;
для второй строки — $TT_{4,3,1,2} = 0 + 0 + 3 + 9 = 12$ и $TT_{4,1,3,2} = 0 + 0 + 3 + 9 = 12$.

Таким образом, кратчайший гамильтонов путь в графе (рис. 2, *в*) включает последовательность работ $J_4\ J_3\ J_1\ J_2$. Как следует из примера, формируемые алгоритмом на рангах текущие пути имеют одинаковые длины. Обоснуем целесообразность применения правил доминирования для выбора лучшего из них.

Обоснование целесообразности использования правил доминирования в базовом алгоритме [21]. Введем следующие определения и предположения.

Определение 1. Текущей последовательностью работ $\sigma(R)$ на ранге R с длиной расписания $T_\sigma(R)$ из частной последовательности работ $\sigma = j_1, j_2, \ldots, j_n$ является последовательность, формируемая на шаге, соответствующем рангу R, и состоящей из работ, выбранных алгоритмом для ранга $R – 1$, и работы на ранге R.

Определение 2. Текущую последовательность работ $\sigma(R)$ на ранге R с длиной расписания $T_\sigma(R)$ из частной последовательности работ $\sigma = j_1, j_2, \ldots, j_n$ назовем «лучшей», чем текущая последовательность $T'_\sigma(R)$, если выполняется условие $T_\sigma(R) - T'_\sigma(R) < 0$ $(T_\sigma(R) < T'_\sigma(R))$.

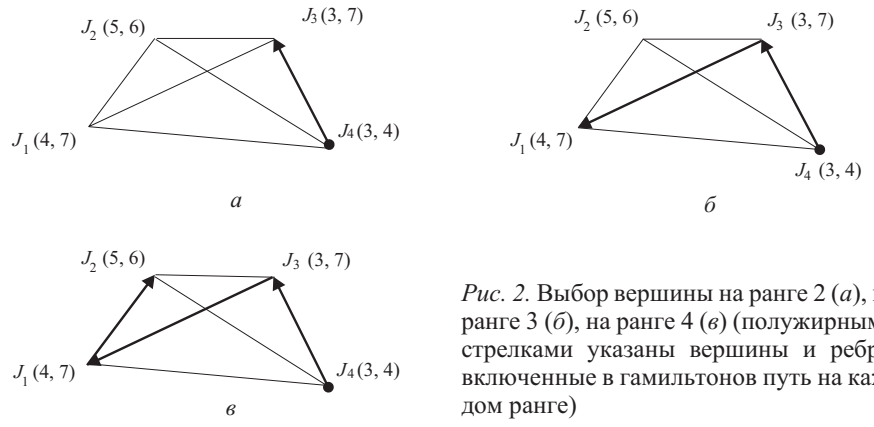

Рис. 2. Выбор вершины на ранге 2 (*а*), на ранге 3 (*б*), на ранге 4 (*в*) (полужирными стрелками указаны вершины и ребра, включенные в гамильтонов путь на каждом ранге)

Определение 3. Текущую последовательность работ $\sigma(R)$ на ранге R с длиной расписания $T_\sigma(R)$ из частной последовательности работ $\sigma = j_1, j_2, ..., j_n$ назовем «эквивалентной» текущей последовательности $T'_\sigma(R)$, если выполняется условие $T_\sigma(R) - T'_\sigma(R) = 0$ $(T_\sigma(R) = T'_\sigma(R))$.

Определение 4. Для работ J_i и J_k, запланированных на предыдущем по отношению к текущему ранге, при условии $T_\sigma(R) - T'_\sigma(R) = 0$ (определение 3) оптимальным является то расписание, в котором J_i предшествует J_k по типу доминирования [24]:

отношение глобального предшествования — работы располагаются в определенном порядке, который определяется по принятым правилам доминирования (невзвешенный и взвешенный случаи);

отношение безусловного предшествования — работы располагаются в таком порядке, при котором их перестановка увеличивает критерий или определяет возможность применения определенного правила доминирования;

отношение предшествования, зависимого от времени, — работы располагаются в таком порядке, при котором их перестановка увеличивает критерий или определяет возможность применения определенного правила доминирования и зависит от момента начала обработки.

Пусть $WT_\sigma(R)$ — взвешенное запаздывание текущей последовательности работ на ранге R из частной последовательности работ σ, а $C_{\max}(\sigma)$ — максимальное время завершения работ в частной последовательности σ. Для приведенных далее определений используем результаты работ [25—27].

Определение 5. Если $C_{\max}(\sigma_1) \le C_{\max}(\sigma_2)$ — максимальные значения времени завершения работ в частных последовательностях σ_1 и σ_2, $WT(\sigma_1) \le WT(\sigma_2)$, а σ_1, σ_2 — частные последовательности работ из после-

довательности $\sigma = j_1, j_2, ..., j_n$, то можно в любом допустимом расписании использовать последовательность σ_1 вместо последовательности σ_2 с оценкой «последовательность σ_1 не хуже последовательности σ_2 по времени завершения всех работ».

Определение 6. Если $C_{\max}(\sigma_1) < C_{\max}(\sigma_2)$ и $WT(\sigma_1) < WT(\sigma_2)$, а σ_1, σ_2 — частные последовательности работ из последовательности $\sigma = j_1, j_2, ..., j_n$, то можно в любом допустимом расписании использовать последовательность σ_1 вместо последовательности σ_2 с оценкой «последовательность σ_1 лучше последовательности σ_2 по времени завершения всех работ».

Для оценки получаемых локально-оптимальных решений сформулируем следующие предположения.

Предположение 1. Для работ J_i и J_k, запланированных на произвольном ранге R, расписание, в котором J_i предшествует J_k, является локально-оптимальным, если выполняется неравенство $T_{ik} - T_{ki} \le 0$ ($T_{ik} \le T_{ki}$).

Предположение 2. Для работ J_i и J_k, запланированных на произвольном ранге R, расписание, в котором J_i предшествует J_k, является локально-оптимальным, если выполняется неравенство $WT_{ik} - WT_{ki} \le 0$ ($WT_{ik} \le WT_{ki}$).

Предположение 3. Расписание частной последовательности работ $\sigma = j_1, j_2, ..., j_n$ на произвольном ранге R является (k, R)-оптимальным, если оно включает k ($R \ge k \ge 1$) локально-оптимальных расписаний.

Предположение 4. Расписание частной последовательности работ $\sigma = j_1, j_2, ..., j_n$ $TT(\sigma)$ является (k, n)-оптимальным, если оно включает k ($n \ge k \ge 1$) локально-оптимальных расписаний.

Предположение 5. Расписание частной последовательности работ $\sigma = j_1, j_2, ..., j_n$ $TWT(\sigma)$ является (k^w, n)-оптимальным, если оно включает k^w ($n \ge k^w \ge 1$) локально-оптимальных расписаний.

Таблица 2. **Правила доминирования для задачи суммарного невзвешенного запаздывания**

Правило	Название	Описание	Формула определения приоритета	Тип
EDD	Earliest Due Date	Наиболее ранний директивный срок	$\min\{d_j\}$	Статическое
MDD	Modified Due Date	Модифицированный директивный срок	$\min\{\max(p_j, d_j - t)\}$	Динамическое
SPT	Shortest Processing Time	Наименьшее процессорное время	$\min\{p_j\}$	Статическое
PD	Processing Time (Due Date)	Время выполнения (директивного срока)	$\min\{p_j / d_j\}$	Статическое

Правила доминирования, использованные для невзвешенного запаздывания, приведены в табл. 2 [11, 28—30].

Из табл. 1 следует, что на рангах имеются конкурирующие пути, из которых требуется выбрать только один, случайно или с помощью правила доминирования, обеспечивающего локально-оптимальное решение, или с помощью «стягивания» всех путей к следующему рангу, т.е. использования всех построенных на текущем ранге путей без отсекания на следующем ранге. Если на каком-либо ранге величины запаздывания всех работ равны, используется правило доминирования EDD.

Например, на ранге 3 $T(3)_{4,3,1} = 6 + (4 - 7) = 3$ и $T(3)_{3,4,1} = 6 + (4 - 7) = 3$. Выбираем $T(3)_{4,3,1}$, так как $d_4 < d_3$. Процедура продолжается до тех пор, пока не достигнет ранга 4, на котором определяется величина кратчайшего гамильтонова пути в графе с минимальным значением суммарного запаздывания. Она и будет суммарным запаздыванием TT, т.е. искомым решением задачи. Таким образом, в данном случае оптимальный путь определяется последовательностью работ J_4, J_3, J_1, J_2 с суммарным запаздыванием $TT_{j_4 j_3 j_2 j_1} = 12$. Последовательность построения гамильтонова пути в графе (см. рис. 1) на каждом ранге показана на рис. 2.

Примеры работы алгоритма с использованием и без использования правил доминирования приведены в табл. 3, а в случае полного перебора — в табл. 4.

Анализ полученных результатов, приведенных в табл. 1 и 3, свидетельствует о том, что, используя правила, можно уменьшить суммарное запаздывание работ. При этом использование различных правил приводит к различным результатам работы базового алгоритма. Использование правила доминирования в некоторых случаях позволяет получить точное решение (см. табл. 1, 3, 4).

При экспериментальном исследовании алгоритма в случае взвешенного запаздывания использованы правила, приведенные в табл. 5 [9—11, 26, 27].

Примеры работы алгоритма в случае взвешенного запаздывания с использованием правила доминирования WMDD и без его использования приведены в табл. 6. Для формализации работы алгоритма с использованием правила доминирования рассмотрим последовательность его реализации с учетом введенных ранее предположений.

Алгоритм минимизации суммарного запаздывания с использованием правила доминирования. Для использования правила доминирования обобщим базовый алгоритм (оптимизации по направлению [21]) с учетом введенных ранее предположений и определений.

Ш а г 1. Формируем произвольную последовательность работ из исходного множества. Эта последовательность составляет ранг 1.

Таблица 3. **Примеры работы алгоритма**

Вершина	Ранг 1	Ранг 2	Ранг 3	Ранг 4
		С использованием правила MDD		
$J_1(4, 7)$	$T(1)_1 =$	$T(2)_{2,1} = (5+4)-7 = 2$	$T(3)_{4,2,1} = (8+4)-7 = 5$	$TT_{4,3,2,1} = 0+0+5+8 = 13$
	$= (0+4)-7 = -3$	$T(2)_{3,1} = (3+4)-7 = 0$	$\boldsymbol{T(3)_{4,3,1} = (6+4)-7 = 3}$	
		$\boldsymbol{T(2)_{4,1} = (3+4)-7 = 0}$	$T(3)_{3,4,1} = (6+4)-7 = 3$	
$J_2(5, 6)$	$T(1)_2 =$	$T(2)_{1,2} = (4+5)-6 = 3$	$T(3)_{4,1,2} = (7+5)-6 = 6$	$\boldsymbol{TT_{4,3,1,2} = 0+0+3+9 = 12}$
	$= (0+5)-6 = -1$	$T(2)_{3,2} = (3+5)-6 = 2$	$\boldsymbol{T(3)_{4,3,2} = (6+5)-6 = 5}$	$TT_{4,1,3,2} = 0+0+3+9 = 12$
		$\boldsymbol{T(2)_{4,2} = (3+5)-6 = 2}$	$T(3)_{3,4,2} = (6+5)-6 = 5$	
$J_3(3, 7)$	$T(1)_3 =$	$T(2)_{1,3} = (4+3)-7 = 0$	$\boldsymbol{T(3)_{4,1,3} = (7+3)-7 = 3}$	
	$= (0+3)-7 = -4$	$T(2)_{2,3} = (5+3)-7 = 1$	$T(3)_{4,2,3} = (8+3)-7 = 4$	
		$\boldsymbol{T(2)_{4,3} = (3+3)-7 = -1}$		
$J_4(3, 4)$	$T(1)_4 =$	$T(2)_{1,4} = (4+3)-4 = 3$		
	$= (0+3)-4 = -1$	$T(2)_{2,4} = (5+3)-4 = 4$		
		$\boldsymbol{T(2)_{3,4} = (3+3)-4 = 2}$		
		Без правила доминирования		
$J_1(4, 7)$	$T(1)_1 =$	$T(2)_{2,1} = (5+4)-7 = 2$	$T(3)_{3,2,1} = (8+4)-7 = 5$	$TT_{4,3,2,1} = 0+0+5+8 = 13$
	$= (0+4)-7 = -3$	$\boldsymbol{T(2)_{3,1} = (3+4)-7 = 0}$	$T(3)_{4,3,1} = (6+4)-7 = 3$	
		$T(2)_{4,1} = (3+4)-7 = 0$	$\boldsymbol{T(3)_{3,4,1} = (6+4)-7 = 3}$	
$J_2(5, 6)$	$T(1)_2 =$	$T(2)_{1,2} = (4+5)-6 = 3$	$T(3)_{3,1,2} = (7+5)-6 = 6$	$TT_{3,4,1,2} = 0+0+3+9 = 14$
	$= (0+5)-6 = -1$	$\boldsymbol{T(2)_{3,2} = (3+5)-6 = 2}$	$\boldsymbol{T(3)_{4,3,2} = (6+5)-6 = 5}$	$TT_{3,1,4,2} = 0+0+6+9 = 15$
		$T(2)_{4,2} = (3+5)-6 = 2$	$T(3)_{3,4,2} = (6+5)-6 = 5$	
$J_3(3, 7)$	$T(1)_3 =$	$T(2)_{1,3} = (4+3)-7 = 0$		
	$= (0+3)-7 = -4$	$T(2)_{2,3} = (5+3)-7 = 1$		
		$\boldsymbol{T(2)_{4,3} = (3+3)-7 = -1}$		
$J_4(3, 4)$	$T(1)_4 =$	$T(2)_{1,4} = (4+3)-4 = 3$	$\boldsymbol{T(3)_{3,1,4} = (7+3)-4 = 6}$	
	$= (0+3)-4 = -1$	$T(2)_{2,4} = (5+3)-4 = 4$	$T(3)_{3,2,4} = (8+3)-4 = 7$	
		$\boldsymbol{T(2)_{3,4} = (3+3)-4 = 2}$		

Ш а г 2. Строим все пути ранга 2. На текущем ранге для каждой работы выбираем пути с минимальной величиной текущего запаздывания. Если их несколько, то выбираем тот путь, который является доминирующим (строго доминирующим), т.е. включает оптимальную последовательность работ на данном ранге.

Ш а г 3. Строим все пути ранга 3 и последующих рангов до тех пор, пока не достигнем ранга n. Рассчитываем суммарное время запаздывания работ для частной последовательности $\sigma = j_1, j_2, ..., j_n$ и суммарное число

Таблица 4. **Пример работы алгоритма в случае полного перебора**

Вершина	Ранг 1	Ранг 2	Ранг 3	Ранг 4	*TT*
$J_1(4,7)$	$T(1)_1 = (0+4)-7 = -3$	$T(2)_{1,2} = 3$	$T(3)_{1,2,3} = 5$	$T(4)_{1,2,3,4} = 11$	$TT_{1,2,3,4} = 19$
			$T(3)_{1,2,4} = 8$	$T(4)_{1,2,4,3} = 8$	$TT_{1,2,4,3} = 19$
		$T(2)_{1,3} = 0$	$T(3)_{1,3,2} = 6$	$T(4)_{1,3,2,4} = 11$	$TT_{1,3,2,4} = 17$
			$T(3)_{1,3,4} = 6$	$T(4)_{1,3,4,2} = 9$	$TT_{1,3,4,2} = 15$
		$T(2)_{1,4} = 3$	$T(3)_{1,4,2} = 6$	$T(4)_{1,4,2,3} = 8$	$TT_{1,4,2,3} = 17$
			$T(3)_{1,4,3} = 3$	$T(4)_{1,4,3,2} = 9$	$TT_{1,4,3,2} = 15$
$J_2(5,6)$	$T(1)_2 = (0+5)-6 = -1$	$T(2)_{2,1} = 2$	$T(3)_{2,1,3} = 5$	$T(4)_{2,1,3,4} = 11$	$TT_{2,1,3,4} = 18$
			$T(3)_{2,1,4} = 8$	$T(4)_{2,1,4,3} = 8$	$TT_{2,1,4,3} = 18$
		$T(2)_{2,3} = 1$	$T(3)_{2,3,1} = 5$	$T(4)_{2,3,1,4} = 11$	$TT_{2,3,1,4} = 17$
			$T(3)_{2,3,4} = 7$	$T(4)_{2,3,4,1} = 8$	$TT_{2,3,4,1} = 16$
		$T(2)_{2,4} = 4$	$T(3)_{2,4,1} = 5$	$T(4)_{2,4,1,3} = 8$	$TT_{2,4,1,3} = 17$
			$T(3)_{2,4,3} = 4$	$T(4)_{2,4,3,1} = 8$	$TT_{2,4,3,1} = 16$
$J_3(3,7)$	$T(1)_3 = (0+3)-7 = -4$	$T(2)_{3,1} = 0$	$T(3)_{3,1,2} = 6$	$T(4)_{3,1,2,4} = 11$	$TT_{3,1,2,4} = 17$
			$T(3)_{3,1,4} = 6$	$T(4)_{3,1,4,2} = 9$	$TT_{3,1,4,2} = 15$
		$T(2)_{3,2} = 2$	$T(3)_{3,2,1} = 5$	$T(4)_{3,2,1,4} = 11$	$TT_{3,2,1,4} = 18$
			$T(3)_{3,2,4} = 7$	$T(4)_{3,2,4,1} = 8$	$TT_{3,2,4,1} = 17$
		$T(2)_{3,4} = 2$	$T(3)_{3,4,1} = 3$	$T(4)_{3,4,1,2} = 9$	$TT_{3,4,1,2} = 14$
			$T(3)_{3,4,2} = 5$	$T(4)_{3,4,2,1} = 8$	$TT_{3,4,2,1} = 15$
$J_4(3,4)$	$T(1)_4 = (0+3)-4 = -1$	$T(2)_{4,1} = 0$	$T(3)_{4,1,2} = 6$	$T(4)_{4,1,2,3} = 8$	$TT_{4,1,2,3} = 14$
			$T(3)_{4,1,3} = 3$	$T(4)_{4,1,3,2} = 9$	$\boldsymbol{TT_{4,1,3,2} = 12}$
		$T(2)_{4,2} = 2$	$T(3)_{4,2,1} = 5$	$T(4)_{4,2,1,3} = 8$	$TT_{4,2,1,3} = 15$
			$T(3)_{3,1,2} = 4$	$T(4)_{4,2,3,1} = 8$	$TT_{4,2,3,1} = 14$
		$T(2)_{4,3} = -1$	$T(3)_{4,3,1} = 3$	$T(4)_{4,3,1,2} = 9$	$\boldsymbol{TT_{4,3,1,2} = 12}$
			$T(3)_{3,2,1} = 5$	$T(4)_{4,3,2,1} = 8$	$TT_{4,3,2,1} = 13$

локально-оптимальных расписаний, т.е. работ, построенных алгоритмом на каждом ранге по определенному правилу доминирования. Полученная последовательность работ σ_1 является строго оптимальной и доминирующей, если для любых частных последовательностей σ_1 и σ_2 на различных рангах

Таблица 5. **Правила доминирования при суммарном взвешенном запаздывании**

Правило	Название	Описание правила	Формула определения приоритета
WEDD	Weighted Earliest Due Date	Взвешенный наиболее ранний директивный срок	$\max\left\{\frac{w_j}{d_j}\right\}$
WMDD	Weighted Modified Due Date	Взвешенный модифицированный директивный срок	$\min\left\{\frac{\max(p_j, d_j - t)}{w_j}\right\}$
WSPT	Weighted Shortest Processing Time	Взвешенное наименьшее процессорное время	$\min\left\{\frac{p_j}{w_j}\right\}$ или $\max\left\{\frac{w_j}{p_j}\right\}$
WPD	Weighted Processing Time Due Date	Отношение взвешенного времени выполнения к директивному сроку	$\max\left\{\frac{w_j}{p_j d_j}\right\}$
WMS	Weighted Minimum Slack	Взвешенный минимальный резерв	$\min\left\{\frac{\max(d_i - p_i - t, 0)}{w_j}\right\}$
CPRT WT	Combined Priority Rule for Total Weighted Tardiness	Комбинированное приоритетное правило для суммарного запаздывания	$\max\limits_{\{J_j \in A\}}\left\{\sum\limits_{\{J_l \in A\}} c_{j,l}(t)\right\}$
ATC	Apparent Tardiness Cost	Очевидная стоимость запаздывания	$\max\limits_{\{J_j \in A\}}\left\{\pi_j = \right.$ $\left. = \frac{w_j}{p_j}\exp\left(\frac{-\max(0, d_j - t - p_j)}{k\,\bar{p}}\right)\right\}$, $k = 2,\ \bar{p} = 1/\lvert A\rvert \sum\limits_{\{J_j \in A\}} p_l$
X-RM	X-dispatch ATC	Модифицированное ATC	$\max\limits_{\{J_j \in A\}}\left\{\pi_j\left(1 - \frac{B\max(0, r_j - t)}{\tilde{p}}\right)\right\}$, $B \in \{1.6, 2\},\ \tilde{p} = 1/\lvert A\rvert \sum\limits_{J_l \in A} p_l$ или $\tilde{p} = \min\limits_{\{J_l \in A\}} p_l$
COVERT	Weighted Cost Over Time	Взвешенная стоимость запаздывания за определенное время	$\max\limits_{\{J_j \in A\}}\left(\frac{w_j}{p_j}\max\left[0, 1 - \frac{\max(0, d_j - t - p_j)}{k\,p_j}\right]\right)$

Таблица 6. **Примеры работы алгоритма в случае взвешенного запаздывания**

Вершина	Ранг 1	Ранг 2	Ранг 3	Ранг 4
		С использованием правила WMDD		
$J_1(4,7,2)$	$WT(1)_1 = ((0+4)-7)*2 = -6$	$WT(2)_{2,1} = ((5+4)-7)*2 = 4$	$WT(3)_{3,2,1} = ((8+4)-7)*2 = 10$	$\boldsymbol{TWT_{4,3,2,1} = 0+0+5\cdot 4+8\cdot 2 = 36}$
		$\boldsymbol{WT(2)_{3,1} = ((3+4)-7)*2 = 0}$	$\boldsymbol{WT(3)_{4,3,1} = ((6+4)-7)*2 = 6}$	
		$WT(2)_{4,1} = ((3+4)-7)*2 = 0$	$WT(3)_{3,4,1} = ((6+4)-7)*2 = 6$	
$J_2(5,6,4)$	$WT(1)_2 = ((0+5)-6)*4 = -4$	$WT(2)_{1,2} = ((4+5)-6)*4 = 12$	$WT(3)_{3,1,2} = ((7+5)-6)*4 = 24$	$TWT_{4,3,1,2} = 0+0+3\cdot 2+9\cdot 4 = 42$
		$\boldsymbol{WT(2)_{3,2} = ((3+5)-6)*2 = 8}$	$\boldsymbol{WT(3)_{4,3,2} = ((6+5)-6)*4 = 20}$	$TWT_{3,1,4,2} = 0+0+3\cdot 5+9\cdot 4 = 51$
		$WT(2)_{4,2} = ((3+5)-6)*4 = 8$	$WT(3)_{3,4,2} = ((6+5)-6)*4 = 20$	
$J_3(3,7,5)$	$WT(1)_3 = ((0+3)-7)*5 = -20$	$WT(2)_{1,3} = ((4+3)-7)*5 = 0$		
		$WT(2)_{2,3} = ((5+3)-7)*5 = 5$		
		$\boldsymbol{WT(2)_{4,3} = ((3+3)-7)*5 = -5}$		
$J_4(3,4,3)$	$WT(1)_4 = ((0+3)-4)*3 = -3$	$WT(2)_{1,4} = ((4+3)-4)*3 = 9$	$\boldsymbol{WT(3)_{3,1,4} = ((7+3)-4)*3 = 18}$	
		$WT(2)_{2,4} = ((5+3)-4)*3 = 12$	$WT(3)_{4,2,3} = ((8+3)-7)*5 = 20$	
		$\boldsymbol{WT(2)_{3,4} = ((3+3)-4)*3 = 6}$		
		Без правила доминирования		
$J_1(4,7,2)$	$WT(1)_1 = ((0+4)-7)*2 = -6$	$WT(2)_{2,1} = ((5+4)-7)*2 = 4$	$WT(3)_{4,2,1} = ((8+4)-7)*2 = 10$	$\boldsymbol{TWT_{4,3,2,1} = 0+6+20+8\cdot 2 = 42}$
		$WT(2)_{3,1} = ((3+4)-7)*2 = 0$	$WT(3)_{4,3,1} = ((6+4)-7)*2 = 6$	
		$\boldsymbol{WT(2)_{4,1} = ((3+4)-7)*2 = 0}$	$\boldsymbol{WT(3)_{3,4,1} = ((6+4)-7)*2 = 6}$	
$J_2(5,6,4)$	$WT(1)_2 = ((0+5)-6)*4 = -4$	$WT(2)_{1,2} = ((4+5)-6)*4 = 12$	$WT(3)_{4,1,2} = ((7+5)-6)*4 = 24$	
		$WT(2)_{3,2} = ((3+5)-6)*4 = 8$	$WT(3)_{4,3,2} = ((6+5)-6)*4 = 20$	$TWT_{3,4,1,2} = 0+6+6+9\cdot 4 = 48$
		$\boldsymbol{WT(2)_{4,2} = ((3+5)-6)*4 = 8}$	$\boldsymbol{WT(3)_{3,4,2} = ((6+5)-6)*4 = 20}$	$TWT_{4,1,3,2} = 0+0+15+9\cdot 4 = 51$

Продолжение табл. 6

Вершина	Ранг 1	Ранг 2	Ранг 3	Ранг 4
$J_3(3, 7, 5)$	$WT(1)_3 = ((0+3)-7)*5 = -20$	$WT(2)_{1,3} = ((4+3)-7)*5 = 0$ $WT(2)_{2,3} = ((5+3)-7)*5 = 5$ $\boldsymbol{WT(2)_{4,3} = ((3+3)-7)*5 = -5}$	$\boldsymbol{WT(3)_{4,1,3} = ((7+3)-7)*5 = 15}$ $WT(3)_{4,2,3} = ((8+3)-7)*5 = 20$	
$J_4(3, 4, 3)$	$WT(1)_4 = ((0+3)-4)*3 = -3$	$WT(2)_{1,4} = ((4+3)-4)*3 = 9$ $WT(2)_{2,4} = ((5+3)-4)*3 = 12$ $\boldsymbol{WT(2)_{3,4} = ((3+3)-4)*3 = 6}$		

выполняются условия: $TT(\sigma_1) < TT(\sigma_2)$ и $k_1 > k_2$ — для невзвешенного запаздывания; $TWT(\sigma_1) < TWT(\sigma_2)$ и $k_1^w > k_2^w$ — для взвешенного запаздывания.

Результаты экспериментального исследования алгоритма с использованием правил доминирования. Для исследования базового алгоритма [21] с использованием правил доминирования разработан программный продукт, с помощью которого рассчитаны полученные в результате экспериментов зависимости времени выполнения (работы алгоритма) и суммарного запаздывания алгоритма для невзвешенного и взвешенного запаздывания (рис. 3—5) на основе методики, предложенной в [23] и алгоритма генерации тестовых последовательностей, приведенного в библиотеке OR-Library [31]. При этом использованы следующие настройки параметров: число экземпляров — 75; диапазон директивных сроков *RDD* = (0,2; 0,4; 0,6; 0,8; 1,0); фактор запаздывания *TF* = (0,2; 0,4; 0,6; 0,8; 1,0); число работ — 160; длительности работ сгенерированы по равномерному закону в интервале [1, 10].

Результаты расчетов, приведенные на рис. 3, подтверждают полученную в [21] оценку временной сложности исследуемого алгоритма. Результаты, представленные на рис. 4, свидетельствуют о том, что при использовании доминирующего правила суммарное время запаздывания уменьшается, и при этом лучшим расписанием является то, которое получено с использованием правила EDD.

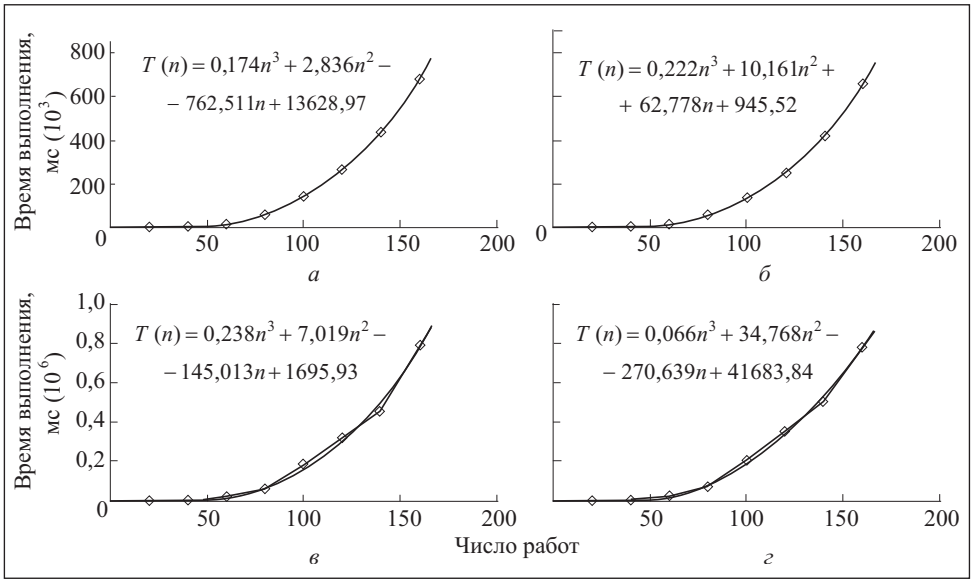

Рис. 3. Зависимость времени выполнения от числа работ для базового алгоритма с использованием метода оптимизации: *а* — EDD; *б* — MDD; *в* — WEDD; *г* — WMDD; *а*, *б* — невзвешенное запаздывание; *в*, *г* — взвешенное запаздывание; ◊ — время выполнения; — аппроксимирующий полином $T(n)$

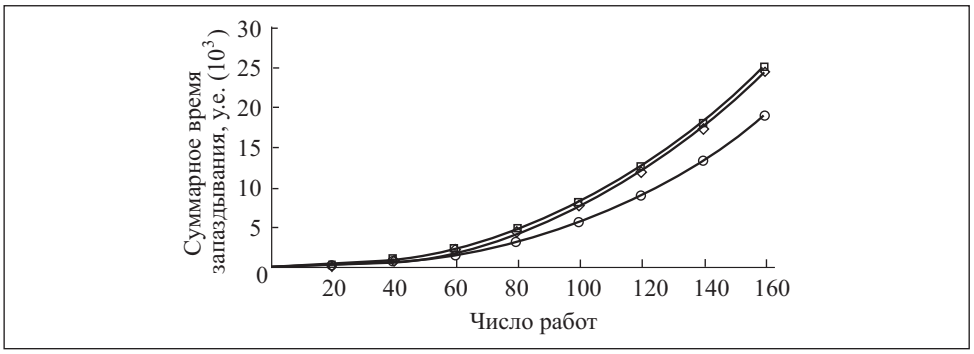

Рис. 4. Зависимость суммарного времени запаздывания от числа работ при использовании базового алгоритма с различными правилами доминирования: ◊ — оптимизация по направлению; ○ — EDD; — MDD; □ — SPT

Метрики оценивания результатов работы алгоритма при использовании правила доминирования. Для оценки улучшения результатов работы базового алгоритма при использовании правила доминирования предлагаются следующие метрики:

отношение числа вершин построенного кратчайшего гамильтонова пути, полученного с использованием правила доминирования, к общему

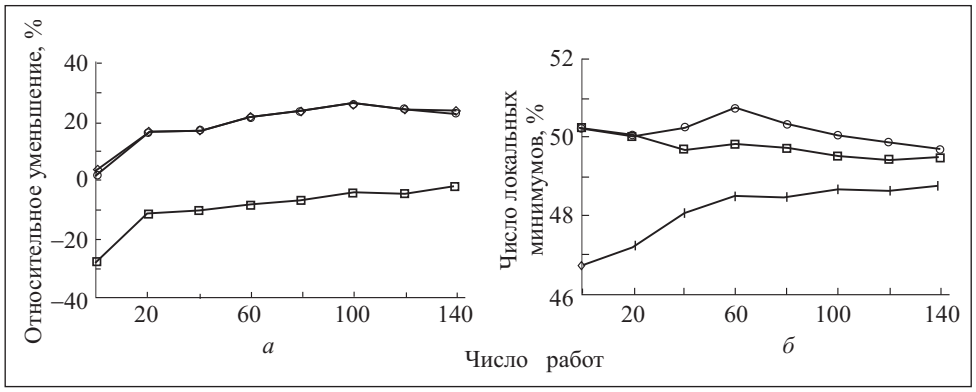

Рис. 5. Зависимость относительного уменьшения суммарного времени запаздывания (*а*) и числа локально-оптимальных решений (*б*) от числа работ при использовании различных методов оптимизации: ○ — EDD; | — MDD; □ — SPT

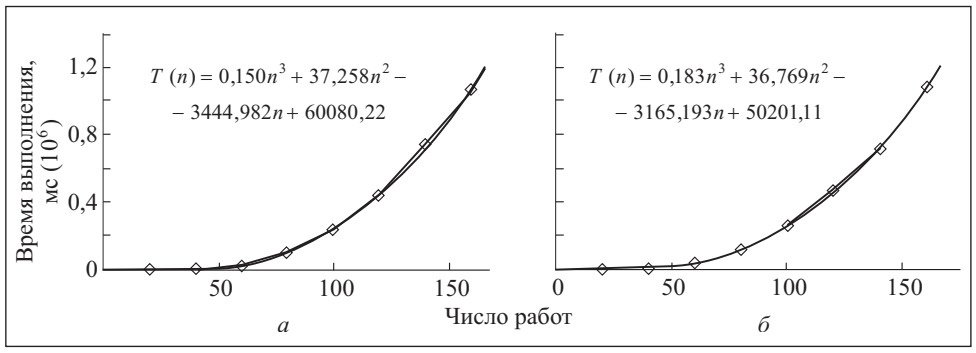

Рис. 6. Зависимость времени выполнения от числа работ для алгоритма оптимизации по направлению без использования правил доминирования (*а*) и с использованием правил EDD (*б*) при мягких директивных сроках: ◊ — время выполнения; —— $T(n)$

числу вершин в графе — плотность $\sum_{GP} v_{dr} \Big/ n$, где v_{dr} — множество вершин графа, полученных на основе определенного правила доминирования;

относительное уменьшение суммарного запаздывания $(TT - TT_{dr})/TT$, где TT_{dr} и TT — суммарное запаздывание с использованием правила доминирования и без его использования.

Для экспериментального исследования работы базового алгоритма с использованием правил доминирования проведены вычислительные эксперименты с исходными данными о количестве практически применяемых работ. Для этого сгенерированы пакеты из 75 заданий по 20 —160 работ в каждом и рассчитаны относительное уменьшение суммарного времени запаздывания и число получаемых локально-оптимальных решений (см. рис. 5).

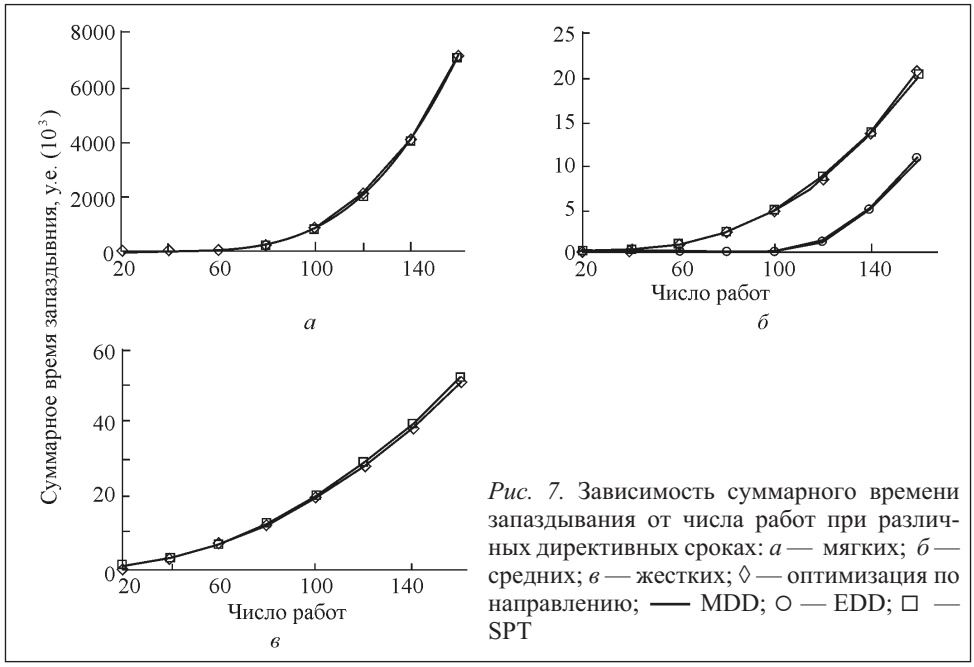

Рис. 7. Зависимость суммарного времени запаздывания от числа работ при различных директивных сроках: *а* — мягких; *б* — средних; *в* — жестких; ◊ — оптимизация по направлению; —— MDD; ○ — EDD; □ — SPT

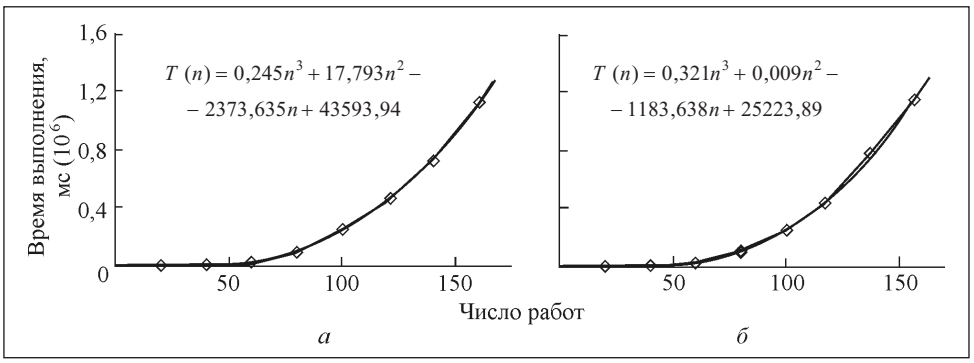

Рис. 8. Зависимость времени выполнения от числа работ при использовании алгоритма оптимизации по направлению без использования правил доминирования (*а*) и с использованием правил COVERT (*б*) для мягких директивных сроков: ◊ — время выполнения; —— $T(n)$

Как следует из рис. 5, *а*, относительное уменьшение суммарного запаздывания при использовании правил достигает 30 %, и лучшие расписания получаются при использовании правил EDD и MDD. Результаты, приведенные на рис. 5, *б*, свидетельствуют о том, что число локально-оптимальных решений, получаемых с использованием правил, достигает 50 %, и лучшим при этом является расписание, получаемое с использованием правила EDD.

Исследование влияния директивных сроков работ на время выполнения алгоритма с использованием правила доминирования. При решении теоретических и практических задач, связанных с планированием работ в распределенных вычислительных системах, важным является определение наиболее точного времени их выполнения. Если в используемых моделях работ учитывается директивный срок, то необходимо учитывать и его величины, так как для работы планировщиков требуется информация о

Таблица 7. **Относительное уменьшение суммарного запаздывания для различных длительностей работ и директивных сроков при использовании правил доминирования**

Диапазон длительности работ, у.е.	*TF*	*RDD*	Суммарное время запаздывания, у.е.				Относительное улучшение, %		
			Без правила	EDD	MDD	SPT	EDD	MDD	SPT
1—10	0,2	1,0	7199,733	0	0	7107,827			1,28
	0,6	0,6	20336,48	10810,27	10578,36	20108,91	46,84	47,98	1,12
	1,0	0,2	51923,33	51143,64	51111,84	52157,99	1,50	1,56	–0,45
10—100	0,2	1,0	71390,69	0	0	75364,17			–5,57
	0,6	0,6	225281,8	123691,9	121989,5	228997,5	45,09	45,85	–1,65
	1,0	0,2	576678,3	572597,8	572070,1	581796,4	0,71	0,80	–0,89
100—1000	0,2	1,0	727745,4	0	0	766290			–5,30
	0,6	0,6	2318171	1272431	1238497	2294737	45,11	46,57	1,01
	1,0	0,2	5748017	5709203	5701751	5801825	0,68	0,80	–0,94

Таблица 8. **Относительное уменьшение суммарного взвешенного запаздывания при**

Диапазон длительности работ, у.е.	*TF*	*RDD*	Суммарное время запаздывания, у.е.					
			Без правила	WEDD	WMDD	WSPT	WPD	COVERT
1—10	0,2	1,0	33709,05	5126,867	0	20239,2	5286,64	4508,347
	0,6	0,6	109812,9	48933,89	41866,95	81906,45	55057,16	59868,28
	1,0	0,2	341932,8	326996,4	326294	332829,3	327690,1	328879,4
10—100	0,2	1,0	378100,6	56600,53	0	194523,3	58576,03	41506,89
	0,6	0,6	1226679	560760,4	479471,2	924313,1	636130,9	716528,1
	1,0	0,2	3776079	3616856	3620074	3691123	3638510	3638459
100—1000	0,2	1,0	3849057	545015,9	0	1954127	615005	403736,6
	0,6	0,6	12310788	5640129	4877216	9440556	6388686	7019181
	1,0	0,2	37838717	36395216	36348787	37038532	36619082	36594292

времени выполнения и сроках завершения работ [32]. Эти величины оцениваются с помощью предполагаемого (ожидаемого) времени, т.е. предварительно рассчитываются пользователями на основании хронологических данных об уже выполненных работах.

В работах [33—35] для определения типов директивных сроков используются типы, определяемые как жесткий, средний и слабый. В данном исследовании жесткость директивного срока определим как его отношение к длительности работы: чем эта величина меньше, тем более жестким является директивный срок. Учитывая, что в генераторе тестовых последовательностей OR-Library [31] диапазон задания директивных сроков регулируется параметром *RDD*, для исследования влияния жесткости директивного срока на работу алгоритма с правилом доминирования используем следующее определение.

Определение 7. Директивные сроки тем жестче, чем большее значение имеет параметр *TF* и чем меньшее — параметр *RDD*.

Заданиями с жесткими директивными сроками будем считать такие, для которых $TF = 1{,}0$, $RDD = 0{,}2$, со средними, — для которых $TF = 0{,}6$, $RDD = 0{,}6$ и с мягкими, — для которых $TF = 0{,}2$, $RDD = 1{,}0$.

Для оценивания влияния рассмотренных типов директивных сроков на работу алгоритма в случае невзвешенного запаздывания проведены расчеты суммарного запаздывания для длительностей работ, распределенных по равномерному закону в интервале [1, 10]. Результаты этих расчетов показали, что время работы алгоритма соответствует полученной в работе [21] оценке временной сложности (рис. 6).

использовании различных правил доминирования

		Относительное уменьшение времени запаздывания, %						
ATC	WMS	WEDD	WMDD	WSPT	WPD	COVERT	ATC	WMS
0	0	84,79		39,96	84,32	86,63		100,00
43137,45	58378,11	55,44	61,87	25,41	49,86	45,48	60,72	46,84
325869,3	338015,1	4,37	4,57	2,66	4,17	3,82	4,70	1,15
0	0	85,03		48,55	84,51	89,02		100,00
498042,4	683777,9	54,29	60,91	24,65	48,14	41,59	59,40	44,26
3607753	3752082	4,22	4,13	2,25	3,64	3,64	4,46	0,64
0	0	85,84		49,23	84,02	89,51		100,00
5020131	6765935	54,19	60,38	23,31	48,10	42,98	59,22	45,04
36291279	37698080	3,81	3,94	2,11	3,22	3,29	4,09	0,37

Графики зависимости времени суммарного запаздывания от числа работ при их длительностях, равномерно распределенных в интервале [1, 10], при различной жесткости директивных сроков приведены на рис. 7. Из представленных графиков видно, что с увеличением жесткости директивных сроков суммарное запаздывание работ увеличивается. При этом наиболее значительное уменьшение суммарного запаздывания обеспечивается алгоритмом, в котором используются правила доминирования EDD и MDD для мягких и средних директивных сроков (табл. 7).

Результаты расчетов суммарного взвешенного запаздывания в диапазоне длительностей работ, распределенных по равномерному закону в интервале [1, 10], для разных типов директивных сроков показали, что время работы алгоритма также соответствует полученной в работе [21] оценке временной сложности. На рис. 8 приведены графики зависимости времени работы алгоритма с использованием правила доминирования и без него для мягких директивных сроков от числа работ.

Результаты проведенных экспериментов и относительное уменьшение суммарного взвешенного запаздывания при различных длительностях работ и весов, равномерно распределенных в интервале [1, 10], для базового алгоритма с использованием различных правил доминирования приведены в табл. 8.

Таким образом, как следует из табл. 8, в случае взвешенного запаздывния лучшие расписания выполнения работ получены при использовании правил доминирования WMDD, ATC и WMS, которые позволяют уменьшить суммарное запаздывание для мягких и средних директивных сроков.

Выводы

Применение правил доминирования позволяет на каждом шаге работы алгоритма выбирать лучшую из текущих последовательностей работ.

Включение правил доминирования в базовый алгоритм незначительно увеличивает время его работы.

Использование правила доминирования для рассмотренного алгоритма минимизации суммарного времени запаздывания позволяет получить локально-оптимальные решения и уменьшить время запаздывания для работ со средними и мягкими директивными сроками.

Современные системы планирования выполнения заданий имеют возможность сортировки заданий с различными приоритетами. Это такие планировщики промежуточного слоя Грид-систем, как ARC NorduGrid, и локальные планировщики, например MAUI, в которых возможно использование предложенного алгоритма, имеющего малую временную слож-

ность, для улучшения локальных расписаний выполнения заданий с произвольными директивными сроками в условиях временных или бюджетных ограничений.

The paper deals with the algorithms of minimizing the total lag of works on a single device on the basis of determining the shortest Hamilton path in the arbitrary graph using the rank approach and the dominance rules. The performance metrics of applying the dominance rules based on the evaluation of locally optimal solutions, obtained on different ranks and the values of relative reduction of total lag in relation to the basic algorithm are proposed. The results of computational experiments for the weighted and unweighted cases for the works with different duration and the types of due dates defined in the research are given. The conditions, under which, the proposed algorithms allow improving the schedule of works are defined and the most effective dominance rules are determined.

СПИСОК ЛИТЕРАТУРЫ

1. *Ye Nong, Gel Esma S., Li Xueping, Farley Toni, Lai Ying-Cheng* Web Server QoS Models: Applying Scheduling Rules from Production Planning // Computers & Operations Research. — 2005. — № 32. — P. 1147—1164.
2. *Zafril Rizal M Azmi, Kamalrulnizam Abu Bakar, Abdul Hanan Abdullah, Mohd Shahir Shamsir, Wan Nurulsafawati Wan Manan* Performance Comparison of Priority Rule Scheduling Algorithms Using Different Inter Arrival Time Jobs in Grid Environment // Intern. J. of Grid and Distributed Computing. — 2011. — Vol. 4, No. 3. — P. 61—70.
3. *Singh H., Dr. Kumar S.* Dispatcher Based Dynamic Load Balancing on Web Server System // Ibid. — 2011. — Vol. 4, No. 3. — P. 89—105
4. *Zafril Rizal M Azmi, Kamalrulnizam Abu Bakar , Mohd Shahir Shamsir, Wan Nurulsafawati Wan Manan, Abdul Hanan Abdullah* Scheduling Grid Jobs Using Priority Rule Algorithms and Gap Filling Techniques // Intern. J. of Advanced Science and Technology. — 2011. — Vol. 37. — P. 61—76.
5. *Klusacek D., Rudova H.* Comparison of Multi-criteria Scheduling Techniques. S. Gorlatch, P. Fragopoulou and T. Priol, Editors. —Springer US, 2008. — P. 173—184.
6. *Klusacek D., Rudova H.* Improving QoS in Computational Grids through Schedule-based Approach // In Scheduling and Planning Applications Workshop at the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS 2008). Sydney, Australia. — 2008. [Электронный ресурс]. — Режим доступа: decsai.ugr.es>~lcv/SPARK/08/Papers/paper04.pdf.
7. *Листровой С.В., Минухин С.В.* Модель и подход к планированию распределения ресурсов в гетерогенных ГРИД-системах // Проблемы управления и информатики. Международный научно-технический журнал. — 2012. — № 5. — С. 120—133.
8. *Листровой С.В., Минухин С.В.* Метод решения задач о минимальном вершинном покрытии в произвольном графе и задачи о наименьшем покрытии // Электрон. моделирование. — 2012. — **34**, №1. — С. 29—43.
9. *Koulamas C.* The Total Tardiness Problem: Review and Extensions // Operations Research. — 1994. — Vol. 42. — P. 1025—1041.
10. *Koulamas C.* The Single-machine Total Tardiness Scheduling Problem: Review and Extensions // European J. of Operational Research. — 2010. — Vol. 202, No. 1. — P. 1—7.
11. *Sena T., Suleka J.M., Dileepan Parthasarati* Static Scheduling Research to Minimize Weighted and Unweighted Tardiness: A State-of-the-art Survey //Intern. J. Production Economics. — 2003. — Vol. 83, No. 1. — P. 1—12.

12. *Du J., Leung J.Y.-T.* Minimizing Total Tardiness on One Machine is NP-hard //Mathematics of Operations Research. — 1990. — № 15. — P. 483—495.
13. *Lawler E.L.* A «Pseudo Polynomial» Algorithm for Sequencing Jobs to Minimize Total Tardiness // Annals of Discrete Mathematics. — 1977. — No1. — P. 331—342.
14. *Baker K.R., Shrage L.E.* Finding an Optimal Sequence by Dynamic Programming an Extension to Precedence-related Tasks // Operations Research Letters. — 1978. — No 26. — P. 111 — 120.
15. *Lawler E.L.* A Fully Polynomial Approximation Scheme for the Total Tardiness Problem // Ibid. — 1982. — No 1. — P. 207—208.
16. *Potts C.N., Van Wassenhove L.N.* Dynamic Programming and Decomposition Approaches for the Single Machine Total Tardiness Problem // European J. of Operational. — 1987. — No 32. — P. 405—414.
17. *Kovalyov M.Y.* Improving the Complexities of Approximation Algorithms for Optimization problems // Operations Research Letters.— 1995.—Vol. 17, March.—P. 85—87.
18. *Koulamas C.* A Faster Fully Polynomial Approximation Scheme for the Single Machine Total Tardiness Problem // European J. of Operational Research.— 2009.— No 193.—P. 637—638.
19. *Павлов А.А., Мисюра Е.Б.* Новый подход к решению задачи «Минимизация суммарного взвешенного опоздания при выполнении независимых заданий с директивными сроками одним прибором» // Системные исследования и информационные технологии. — 2002. — № 2. — С. 7—32.
20. *Павлов О.А., Мисюра Е.Б., Шевченко К.Ю.* Побудова ПДС-алгоритму розв'язання задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі // Вісн. НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. — Київ: Век+, 2012. — № 56. — С. 58—70.
21. *Мінухін С.В.* Метод мінімізації часу виконання завдань з директивними строками на некластеризованому ресурсі обчислювальної системи // Інформаційно-керуючі системи на залізничному транспорті. — 2009. — № 3. — С. 47 — 53.
22. *Minukhin S.* Efficient Method for Single Machine Total Tardiness Problem // IV Intern. Conf. «Problems of Cybernetics and Informatics» (PCI'2012), September 12—14, 2012. [Электронный ресурс].— Режим доступа: www.pci2012.science.az/1/23.pdf.
23. *Мінухін С.В., Лєнько Д.С.* Дослідження методів мінімізації сумарного часу запізнювання робіт з директивними строками на одиночному обчислювальному ресурсі // Проблеми і перспективи розвитку ІТ-індустрії. Системи обробки інформації. — 2013. — Вип. № 3 (110). Т. 2. — С. 30—35.
24. *Шевченко К.Ю.* Алгоритм гілок та меж для статистичних досліджень нового ПДС-алгоритму розв'язання задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі // Вісн. НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. — Київ: Век+, — 2012. — № 55. — С. 56—69.
25. *Chu C.* Efficient Heuristics to Minimize Total Flow Time with Release Dates // Operations Research Letters. — 1992. — №12. — P. 321—330.
26. *Jouglet A., Savourey D.D., Carlier J., Baptiste P.* Dominance-Based Heuristics for One-Machine Total Cost Problems // European J. of Operational Research. — 2008. — 184 (3). — P. 879— 899.
27. *Jouglet A., Savourey D.* Dominance Rules for the Parallel Machine Total Weighted Tardiness Scheduling Problem with Release Dates//Computers & Operations research. — 2011. — Vol. 38, Issue 9. — P. 1259—1266.
28. *Jackson J. R.* Scheduling a Production Line to Minimize Maximum Tardiness. Minimizing Total Tardiness for One Machine Research, Report 43, Management Science Research Project, UCLA, 1955. [Электронный ресурс]. — Режим доступа: http://citeseerx.ist.psu.edu/showciting?cid=38200.

29. *Lai Tsung-Chyan, Kuo Yuh-Kwo* Minimizing Total Tardiness for Single Machine Sequencing // J. of the Operations Research Society of Japan.— 1996. — Vol. 39, № 3. — P. 316—321.
30. *Scheduling* Single Machine Scheduling [Электронный ресурс].` — Режим доступа: www.or.uni-bonn.de/lectures/sched10_2.pdf.
31. *OR-Library* [Электронный ресурс]. — Режим доступа: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html.
32. *Zotkin D., Keleher P.J.* Job-Length Estimation and Performance in Backlling Schedulers // In Proc. 8th High Performance Distributed Computing Conf. IEEE, 1999. [Электронный ресурс]. — Режим доступа: http://www.umiacs.umd.edu/~dz/pbpslist/hpdc1999.pdf.
33. *Mohamadreza Kaviani, Majid Aminnayeri, Seyed Nima Rafienejad, Fariborz Jolai.* An Appropriate Pattern to Solving a Parallel Machine Scheduling by a Combination of Meta-heuristic and Data Mining // J. of American Science. — 2012. — № 8 (1). — P. 160—167.
34. *Chandra P., Li S., Stan M.* Jobs and Tool Sequencing in an Automated Manufacturing Environment // Intern. J. of Production Research. — 1993. — Vol. 31, Issue 1. — P. 2911—2925.
35. *Sekhar C.V. S., Devi M.P.* Improvement of Performance in a Maintenance Job Shop // Proc. of the 11th WSEAS Intern. Conf. on Applied Mathematics. Dallas, Texas, USA, March 22—24, 2007. — P. 29—34.

Поступила 19.11.13;
после доработки 17.12.13

МИНУХИН Сергей Владимирович, канд. техн. наук, профессор кафедры информационных систем Харьковского национального экономического университета. В 1976 г. окончил Харьковский ин-т радиоэлектроники. Область научных исследований — оптимизация функционирования распределенных вычислительных систем, распределенные вычисления, облачные вычисления.

ЛЕНЬКО Дмитрий Сергеевич. В 2013 г. окончил Харьковский национальный экономический университет. Область научных исследований — методы и алгоритмы построения расписаний.