Отчет по проекту No.11-08-01321-а "Графический подход решения инженерных задач комбинаторной и дискретной оптимизации"

Представлен графический метод решения задач комбинаторной оптимизации, при решении которых допускается декомпозиция задачи на подзадачи меньшей размерности и использование принципа оптимальности Беллмана при их решении. В отличие от алгоритмов динамического программирования, использующих тот же принцип, в графическом алгоритме все возможные состояния системы рассматриваются не отдельно, а группами. Это становится возможным если принимать во внимание аналитический вид целевой функции, т.е. работать с "графиком" функции, преобразовывая его на каждой стадии согласно аналитическому виду. Графический метод позволяет значительно сократить трудоемкость решения многих задач, а также строить эффективные аппроксимационные схемы. Результаты численных экспериментов свидетельствуют об эффективности графического метода.

Графический метод использует тот же декомпозиционный подход к решению задач комбинаторной оптимизации, что и метод динамического программирования.

Словосочетание "динамическое программирование" впервые было использовано в 1940-х годах Р. Беллманом[1] для описания процесса нахождения решения задачи, где ответ на одну задачу может быть получен только после решения задач, "предшествующих" ей. Ключевая идея метода динамического программирования заключается в следующем. Зачастую, чтобы решить поставленную задачу, можно разбить ее на подзадачи, решить их, также используя декомпозицию, и после чего объединить решения подзадач в одно общее решение. В процессе декомпозиции возникает множество идентичных подзадач, и только одна задача множества решается в методе динамического программирования, что позволяет сократить перебор допустимых решений.

Для краткости будем обозначать алгоритм динамического программирования через *DPA* (Dynamic Programming Algorithm), а графический алгоритм, соответственно, *GrA* (Graphical Algorithm). В DPA на каждом

[1] Метод описан Беллманом в привычной нам терминологии в 1953м году.

шаге (на каждой стадии) j вычисляются значения некоторой функции $F_j(t)$ для каждого возможного значения аргумента t (для каждого состояния) процесса принятия решения, где $t \in [0, A]$, $t \in Z$, где A – некоторый числовой параметр задачи. Будем называть функцию $F_j(t)$ функцией Беллмана. Фактически эта функция соответствует значениям целевой функции для подзадачи размерности j если она решается в условиях (при состоянии системы) t. На практике DPA реализуется в режиме "калькулятора т.е. перебираются все состояния $t \in [0, A]$, $t \in Z$. Для каждого состояния проводятся несложные вычисления, и полученное значение $F_j(t)$ сохраняется в ячейку памяти (в таблицу). Эти $A+1$ числовых значений используются на следующем шаге $j+1$. Однако зачастую нет необходимости вычислять (и сохранять в памяти) значения $F_j(t)$ для каждой точки $t \in [0, A]$, $t \in Z$. Может оказаться, что на некотором интервале $[t_l, t_{l+1})$ вычисляемая функция представима аналитически в виде $F_j(t) = \varphi(t)$ (например, $F_j(t) = k \cdot t + b$, т.е. функция $F_j(t)$ определена в том числе для нецелых значений аргумента t). Тогда процесс вычисления функции $F_{j+1}(t)$ на шаге $j+1$ можно организовать таким образом, чтобы учитывать не отдельные значения $F_j(t)$, а преобразовывать функцию $F_j(t)$ в $F_{j+1}(t)$ аналитически, согласно заданным рекурсивным уравнениям Беллмана.

Пусть для некоторой проблемы минимизации целевого функционала заданы следующие рекурсивные уравнения Беллмана:

$$F_j(t) = \min \left\{ \begin{array}{ll} \Phi^1(t) = \alpha_j^1(t) + F_{j-1}(t - a_j^1), & j = 1, 2, \ldots, n; \\ \Phi^2(t) = \alpha_j^2(t) + F_{j-1}(t - a_j^2), & j = 1, 2, \ldots, n; \\ \ldots & \ldots \\ \Phi^{k_j}(t) = \alpha_j^{k_j}(t) + F_{j-1}(t - a_j^{k_j}), & j = 1, 2, \ldots, n, \end{array} \right. \tag{1}$$

с начальными условиями

$$\begin{array}{lll} F_0(t) = 0, & \text{для} & t \geq 0, \\ F_0(t) = +\infty, & \text{для} & t < 0. \end{array} \tag{2}$$

В (1) значения k, $k = 1, \ldots, k_j$, будем называть "сратегиями" , определенными для данного элемента системы j. Каждой стратегии k, $k = 1, \ldots, k_j$, соответствует функция $\Phi^k(t)$ и значение переменной $x_j = x_j^k$, участвующей в итоговом решении. Переменная x_j – управляемый параметр на шаге j процесса. Трудоемкость DPA для такой системы рекурсивных уравнений равна $O(\sum_{j=1}^n k_j A)$ операций.

Таблица 1: Вычисления в DPA

t	0	1	2	...	y	...	A
$F_j(t)$	знач$_0$	знач$_1$	знач$_2$	...	знач$_y$	...	знач$_A$
частичное оптим. решение $X(t)$	$X(0)$	$X(1)$	$X(2)$	...	$X(y)$	...	$X(A)$

Таблица 2: Вычисления в GrA

t	$[t_0, t_1)$	$[t_1, t_2)$	...	$[t_l, t_{l+1})$	...	$[t_{m_j-1}, t_{m_j}]$
$F_j(t)$	$\varphi_1(t)$	$\varphi_2(t)$	...	$\varphi_{l+1}(t)$	...	$\varphi_{m_j}(t)$
частичное оптим. решение $X(t)$	$X(t_0)$	$X(t_1)$	...	$X(t_l)$	...	$X(t_{m_j-1})$

Для задач с булевыми переменными, таких как задача о ранце или задача разбиения, для некоторых оноприборных задач теории расписаний, $k_j = 2,\ j = 1, \ldots, n$ и $x_j^1 = 1,\ x_j^2 = 0$. Если в таких задач вычисления DPA необходимо провести на каждом шаге $j = 1, 2, \ldots, n$, где n размерность задачи, то трудоемкость DPA обычно равна $O(nA)$ операций. Переменная x_j – для задачи о ранце имеет интерпретацию "включать или не включать предмет в ранец" .

На шаге $j,\ j = 1, 2, \ldots, n$, DPA мы вычисляем и сохраняем в памяти компьютера данные, представленные в Таблице 1.

В Таблице 1 вектор $X(y),\ y = 0, 1, \ldots, A$, состоит из j элементов (значений) $x_1, x_2, \ldots, x_j$. Данный вектор описывает некоторое частичное оптимальное решение.

Та же самая информация иногда может быть представлена в виде, приведенном в Таблице 2, где $0 = t_0 < t_1 < t_2 < \ldots < t_{m_j} = A$. Точки $t_1, \ldots, t_{m_j}$ будем называть *точками излома*, т.к. в них изменяется уравнение, задающее функцию $F_j(t)$.

Для вычисления функции $F_{j+1}(t)$ в GrA мы сравниваем промежуточные функции $\Phi^1(t), \ldots, \Phi^{k_{j+1}}(t)$, которые определяются следующим образом.

Функция $\Phi^k(t)$ является комбинацией двух функций $\alpha_{j+1}^k(t)$ и $F_j(t - a_{j+1}^k)$. Функция $F_j(t - a_{j+1}^k)$ имеет ту же структуру, как представлено в Таблице 2, но где каждый интервал $[t_l, t_{l+1})$ заменен на интервал

$[t_l - a^k_{j+1}, t_{l+1} - a^k_{j+1})$, т.е. мы сдвигаем график функции $F_j(t)$ вправо на a^k_{j+1} единиц. Если аналитический вид функций $F_j(t)$ и $\alpha^k_{j+1}(t)$ несложен (например, они являются кусочно-линейными функциями), то построить функцию их суммы также несложно. Если функция $\alpha^k_{j+1}(t)$ представима в виде Таблицы 2 с μ_k столбцами, тогда можно представить функцию $\Phi^k(t)$ в табличном виде, с $m_j + \mu_k$ столбцами.

Функцию $F_{j+1}(t) = \min\{\Phi^1(t), \ldots, \Phi^{k_{j+1}}(t)\}$ мы вычисляем следующим образом. Пусть таблица, которой задана функция $\Phi^k(t)$, содержит интервалы (столбцы)

$$[t^k_0, t^k_1), [t^k_1, t^k_2), \ldots, [t^k_{(m_j+\mu_k)-1}, t^k_{(m_j+\mu_k)}],\ k = 1, \ldots, k_{j+1}.$$

Для вычисления функции $F_{j+1}(t)$, мы сравниваем все функции $\Phi^k(t),\ k = 1, \ldots, k_j$, на каждом интервале, сформированном точками

$$\{t^k_0, t^k_1, t^1_2, \ldots, t^k_{(m_j+\mu_k)-1}, t^k_{(m_j+\mu_k)} | k = 1, \ldots, k_{j+1}\}.$$

На каждом из этих интервалов уравнение, задающее функцию $\Phi^k(t),\ k = 1, \ldots, k_j$, известно (согласно определению), поэтому можно аналитически найти точки их пересечения (например, если функции – кусочно-линейные, то чтобы найти точки пересечения их линейных фрагментов достаточно решить тривиальное уравнение) и построить функцию минимума за $O(m_j + \mu_k)$ операций, где $m_j + \mu_k$ – количество интервалов, на которых ищутся точки пересечения и сравниваются фрагменты функций. Пусть μ' – общее количество точек пересечения. Тогда в таблице, соответствующей функции $F_{j+1}(t)$, будет не более $M = k_{j+1} m_j + \sum\limits_{i=1}^{k_{j+1}} \mu_i + \mu'$ интервалов. GrA можно построить таким образом, что все нецелочисленные точки излома будут исключены из таблицы $F_{j+1}(t)$. Например, если в таблице присутствуют два интервала $[t_{k-1}, t_k)$ и $[t_k, t_{k+1})$, где $t_k \notin Z$, то можно заменить интервалы на $[t_{k-1}, \lfloor t_k \rfloor]$ и $[\lfloor t_k \rfloor + 1, t_{k+1})$ и учитывать полученный разрыв на последующих шагах. Тогда $M \leq A$. Фактически на каждом шаге $j,\ j = 1, 2, \ldots, n$, графического алгоритма мы рассматриваем не все точки $t \in [0, A]$, $t \in Z$, но только точки, в которых меняется оптимальное частичное решение или меняется аналитический вид функции $F_j(t)$. Для некоторых задач (для некоторых целевых функций), количество таких точек M небольшое, и графический алгоритм имеет трудоёмкость $O(\sum_{j=1}^n k_j \min\{A, M\})$ операций, вместо $O(\sum_{j=1}^n k_j A)$ операций как в алгоритме динамического программирования.

Более того, GrA имеет следующие преимущества перед DPA:

- с помощью GrA можно решать примеры с нецелочисленными параметрами;

- время работы GrA на двух примерах со множеством числовых параметров P и множеством параметров $\{b \cdot 10^l \pm 1 | b \in P\}, l > 1$, совпадает, в то время, как время работы DPA будет в 10^l раз больше на втором примере. То есть с помощью GrA, можно решать примера “большого масштаба”;

- принимаются во внимание внутренние свойства задачи (например, для задачи о ранце, может оказаться, что предмет с наименьшей удельной стоимостью не оказывает влияние на функцию Беллмана);

- известно, что GrA для некоторых задач имеет полиномиальную трудоемкость, в то время, как исходный алгоритм DPA – псевдополиномиальную. Или же GrA существенно сокращает трудоемкость DPA (например, для задачи $1(no-idle)|| \max \sum w_j T_j$).

Построение приближенных графических алгоритмов.

Предложена полная полиномиальная аппрокимационная схема (FPTAS), основанная на данном графическом алгоритме, которая имеет меньшую трудоемкость среди известных аналогичных алгоритмов.

Графический алгоритм может быть модифицирован в FPTAS следующим образом. Пусть мы рассматриваем некоторую задачу минимизации. Пусть $\delta = \frac{\varepsilon LB}{n}$, где LB – некоторая известная нижняя оценка целевой функции. Пусть также известна верхняя оценка UB такая, что $\frac{UB}{LB}$ не превышает некоторую константу c. Чтобы сократить трудоемкость GrA, необходимо сократить количество колонок в таблицах $F_j(t)$. Причем количество этих колонок равно количеству различных значений $0 = b_j^1, b_j^2, b_j^3, \ldots, b_j^{m_j+1}$ в этих колонках. Мы можем рассматривать только те состояния системы (и интервалы), для которых $F_j(t) < UB$, т.к. остальные значения не влияют на итоговую целевую функцию. Тогда примем $b_j^{m_j+1} < UB$.

В таблице $F_j(t)$ мы будем хранить не оригинальные значения b_j^k, но значения $\overline{b_j^k}$ являющиеся ближайшим к b_j^k значениями, делящимися без остатка на δ. Существует не более чем $\frac{UB}{\delta} = \frac{c \cdot n}{\varepsilon}$ различных значений $\overline{b_j^k}$. Тогда можно будет преобразовать таблицу функции $F_j(t)$ в таблицу приближенной функции с не более чем $2\frac{c \cdot n}{\varepsilon}$ колонками. Причем для данной модифицированной функции $F'_j(t)$ выполняется $|F_j(t) - F'_j(t)| < \delta \le \frac{\varepsilon F(\pi^*)}{n}$. Если мы будем выполнять подобную аппроксимацию после каждой стадии GrA, то накопленная ошибка не превысит значения $n\delta \le \varepsilon F(\pi^*)$, и трудоемкость приближенного GrA будет $O(\frac{n^2}{\varepsilon})$, если трудоемкость исходного GrA равна $O(nd_{max})$.

Сравнение графического алгоритма и двух реализаций метода динамического программирования

Сравниваются некоторые свойства графического метода и двух возможных реализаций метода динамического программирования для вышеупомянутых задач. Полученные результаты представлены в следующей таблице. Пусть $\Theta_l = \{x_1p_1 + x_2p_2 + \ldots + x_lp_l | x_1, x_2, \ldots, x_l \in \{0, 1\}\}$ и $t_n^{LB} \in (-\infty, +\infty)$ – значение, при котором выполняется $F_n(t_n^{LB}) = LB$.

Таблица 7: Сравнение DPA, GrA и альтернативного DPA

Свойство	Классич. DPA	GrA	Альтернатив DPA
Может ли решать примеры $p_j \notin Z$ и примеры с большими p_j	нет	да	да
рассмотрены состояния t	все $t \in [0, d_{max}] \bigcap Z$	только t, где изменяется наклон функции $F_j(t)$	только t из Θ_l
Трудоемкость для исходного примера - для задачи $1\|\| \sum GT_j$ - для задачи$1(no\text{-}idle)\|\| \max \sum w_j T_j$ is	$O(n \min\{d_{max}, UB\})$ $O(nd_{max})$ $O(n \min\{d_{max}, UB\})$	$O(n \min\{d_{max}, UB\})$ $O(n \min\{d_{max}, UB\})$ $O(n \min\{d_{max}, UB, \sum w_j\})$	$O(n \min\{d_{max}, UB\})$ $O(n \min\{d_{max}, UB\})$ $O(n \min\{d_{max}, UB\})$
Находит оптимальные решения для всех возможных времен старта обслуживания $t \in [0, d]$ за время	$O(nd_{max})$	$O(nd_{max})$	-
$\ldots t \in (-\infty, t_n^{UB}]$ за время	$O(nUB)$	$O(nUB)$	-
$\ldots t \in (-\infty, +\infty)$ за время	$O(nF(\pi', d_{max}))$	$O(nF(\pi', d_{max}))$	-
Трудоемкость FPTAS	$O(\frac{n^3}{\varepsilon} \log \frac{n}{\varepsilon}))$	$O(n^3/\varepsilon)^*$	$O(n^3/\varepsilon)^{**}$

* За это время, для всех $t \in (-\infty, t_n^{UB}]$ будут найдены решения, абсолютная погрешность которых не превосходит εLB. Для всех $t \in [t_n^{LB}, t_n^{UB}]$, $t_n^{LB} \leq 0 \leq t_n^{UB}$, будут найдены решения с относительной погрешностью, не больше $1 + \varepsilon$.
** Приближенное решение будет найдено только для времени старта обслуживания требований $t = 0$.

Таблица 8: Полученные результаты

Задача	Трудоемкость GrA	Трудоемкость FPTAS	Док-ва NP-труд-ти
$1\|\|\sum w_jU_j$	$O(\min\{2^n, n\cdot\min\{d_{max}, F_{opt}\}\})$		
$1\|d_j=d'_j+T\|\sum U_j$	$O(n^2)$		
$1\|\|\sum GT_j$	$O(\min\{2^n, n\cdot\{d_{max}, nF^*\}\})$	$O(n^2\log\log n + \frac{n^2}{\varepsilon})$	+
$B-1$	$O(\min\{2^n, n\cdot\min\{d_{max}, F^*\}\})$	$O(n^2/\varepsilon)$	+
$B-1G$	$O(\min\{n^2\cdot\min\{d_{max}, F^*\}\})$	$O(n^3/\varepsilon)$	+
$1\|d_j=d\|\sum w_jT_j$	$O(\min\{n^2\cdot\min\{d, F^*\}\})$	$O(n^3/\varepsilon)$	
$1(no\text{-}idle)\|\|\max\sum w_jT_j$	$O(\min\{2^n, n\cdot\min\{d_{max}, nF^*, \sum w_j\}\})$	$O(n^2\log\log n + \frac{n^2}{\varepsilon})$	+
$1(no\text{-}idle)\|\|\max\sum T_j$	$O(n^2)$		
$1(no\text{-}idle)\|r_j\|\max\sum T_j$			+
Задача об инвестициях	$O(\sum k_jA)$	$O(\frac{n\cdot\sum k_j}{\varepsilon}(1+\log\log n))$	
P-cost	$O(nA\log A)$	$O(n^2/\varepsilon\log(n/\varepsilon)+n^2\log\log(\sum_{j:=1}^n c_jb_j))$	+

Публикации по результатам проекта:

Монографии

1. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы // Учебное пособие М:Издательство МГУ, 2011, 223 с.

2. Лазарев А.А., Мусатова Е.Г., Гафаров Е.Р., Кварацхелия А.Г. Теория расписаний. Задачи железнодорожного планирования//Научное издание, М:ИПУ РАН, 2012, 92 с.

3. Лазарев А.А., Мусатова Е.Г., Гафаров Е.Р., Кварацхелия А.Г. Теория расписаний. Управление транспортными системами// Учебное пособие М:Издательство МГУ, 2012, 159 с.

4. A.A. Lazarev and E.R. Gafarov (2011), Scheduling Theory. Single machine tardiness problems. LAP LAMBERT Academic Publishing GmbH & Co. KG, 87 pages. (in Russian).

Статьи в рецензируемых журналах

5. E.R. Gafarov, A. Dolgui (2013), Two Dedicated Machines Scheduling Problem in Two-Sided Assembly Lines. Optimization Letters, Vol. 7. DOI 10.1007/s11590-013-0671-0.

6. E.R. Gafarov, A.Dolgui, A.A. Lazarev and F.Werner (2013), A Graphical Approach to Solve an Investment Optimization Problem. Journal of Mathematical Modelling and Algorithms in Operations Research (в печати).

7. E.R. Gafarov, A.A. Lazarev and F. Werner. (2012), A note on a single machine scheduling problem with generalized total tardiness objective function. Information Processing Letters, 112 (3), 72 - 76. (among the 25 most downloaded papers of this journal in Science Direct within the last 90 days in January 2012) (journal impact factor 2011: 0.455).

8. E.R. Gafarov, A.A. Lazarev and F. Werner (2012), Transforming a pseudo-polynomial algorithm for the single machine total tardiness maximization problem into a polynomial one. Annals of Operations Research, 196 (1), 247-261. (journal impact factor 2011: 0.840).

9. E.R. Gafarov, A.A. Lazarev and F. Werner (2012), Approximability Results for the Resource-Constrained Project Scheduling Problem with a Single Type of Resources. Annals of Operations Research, doi:10.1007/s10479-012-1106-5 (accepted, in print). (journal impact factor 2011: 0.840).

10. E.R. Gafarov, A.A. Lazarev, and F. Werner (2011), Single Machine Scheduling Problems with Financial Resource Constraints: Some Complexity Results and Properties. Mathematical Social Sciences, 62, 7-13. (among the 25 most downloaded papers of this journal in Science Direct from July to September 2011 and also from October to December 2011) (journal impact factor 2011: 0.429).

Тезисы конференций

11. E. R. Gafarov, A. Dolgui, A. Lazarev, SOLUTION ALGORITHMS FOR THE TWO-STATION SINGLE TRACK RAILWAY SCHEDULING PROBLEM / MISTA 2013 PROCEEDINGS. Gent, Belgium: The Belgian Operational Research Society, 2013. pp. 636-640.

12. E. R. Gafarov, A. Dolgui, A. Lazarev, F. Werner, A Graphical Algorithm for Solving an Investment Optimization Problem / MISTA 2013 PROCEEDINGS. Gent, Belgium: The Belgian Operational Research Society, 2013. pp. 290-299.
E. R. Gafarov, A. Dolgui, A. Lazarev, F. Werner, A Graphical Approach for Solving Single Machine Scheduling Problems Approximately / Preprints of the IFAC Conference on Manufacturing Modelling, Management and Control, 2013. pp. 1356-1361.

13.E. R. Gafarov, A. Dolgui, A. Lazarev (2012), Notes on Complexity of the Simple Assembly Line Balancing Problem, Proceedings of the Conference UKI 2012, Institute of Control Sciences of the Russian Academy of Sciences, Moscow, Russia, 16-19 April, pp. 259-266.

14. E.R. Gafarov, A. Dolgui, A.A. Lazarev (2012), Some Complexity Results for the Simple Assembly Line Balancing Problem, Proceedings of the 3rd International Conference Optimization and Applications (OPTIMA 2012), Costa da Caparica, Portugal, 23-30 September 23-30, Edited by V.I. Zubov, Moscow, Russian Academy of Sciences, 2012, ISBN 978–5–91601–051–0, pp. 81-85.

15.E.R. Gafarov, A. Dolgui (2012), Algorithms for the two-station single track railway scheduling problem, Book of abstracts of the International Conference "Information Technologies in Industry" (ITI 2012), Minsk, Belarus, 30 October -1 November, ISBN 978-985-6744-78-8, pp. 111-112.

16.E.R. Gafarov, A.A. Lazarev, F. Werner (2012), Graphical Approach to Solve Combinatorial Problems: Algorithms and Some Computational Results, Proceedings of the 14th IFAC Symposium on Information Control Problems in Manufacturing (INCOM 2012), Bucharest, Romania, 23-25 May, Editors: N. Bakhtadze, A. Dolgui, Elsevier IFAC-PapersOnLine, ISSN 1474-6670, vol. 14, part 1, pp. 127-132.

17.E.R. Gafarov, A.A. Lazarev, F. Werner (2011), Scheduling Problems with Financial Resource Constraints. Proceedings of the 2nd International conference "Optimization and Applications"(Optima 2011), Petrovac, Montenegro, 25 September – 2 October, pp. 82-85.

18. Гафаров Е.Р., Лазарев А.А. (2013), Математические методы оптимизации при составлении учебного расписания, Сборник научных трудов 13-й международной научно-практической конференции "Новые информационные технологии в образовании". М.: ООО "1С-Паблишинг". С. 51-55.

19.Гафаров Е.Р., Лазарев А.А. (2012), Алгоритмы решения для задач теории расписаний на однопутной железной дороге, Труды 1-й научно-технической конференции "Интеллектуальные системы управления на железнодорожном транспорте"(ИСУЖТ-2012, Москва). М.: ОАО "НИИАС". С. 114- 129.

20. E. R. Gafarov, A. Dolgui, A. Lazarev, F. Werner (2013), SOLVING AN INVESTMENT OPTIMIZATION PROBLEM BY AN IMPROVED GRAPHICAL APPROACH, 22nd International Conference on Production Research, Brazil, 28.07-01.08, 6 с.

21. E.R. Gafarov, A. Dolgui, F. Grimaud (2012), Two Dedicated Machines Scheduling Problem in Two-Sided Assembly Lines. Conference Guide Abstracts Book of the conference Operations Research 2012, Hannover, Germany, 4-7 September, p. 174. (to appear in Operations Research Proceedings 2012).

22. Лазарев А.А., Мусатова Е.Г., Гафаров Е.Р. (2013), The problem of train timetable change for the case of repair works, EURO-INRORMS Rome 2013. Rome: Sapienza Universita di Roma. С. 156.

23. E. R. Gafarov, A. Dolgui, A. Lazarev, F. Werner, A Graphical Approach for Solving Single Machine Scheduling Problems Approximately / Preprints of the IFAC Conference on Manufacturing Modelling, Management and Control, 2013. pp. 1356-1361.

Препринты

24. E.R. Gafarov, A. Dolgui (2012), Two-Station Single Track Railway Scheduling Problem With Equal Speed of Trains, Report n° RR-12-09, LIMOS, CNRS UMR 6158, April 7th, 14 pages.

25. E.R. Gafarov, A. Dolgui (2012), Two Customized Parallel Machines Scheduling Problem with Precedence Relations, Report n° RR-12-10, LIMOS, CNRS UMR 6158, April 7th, 9 pages.

26. E.R. Gafarov, A. Dolgui (2012), Hard Special Case and Other Complexity Results for SALBP-1, Report n° RR-12-08, LIMOS, CNRS UMR 6158, April 7th, 13 pages.

27. E.R. Gafarov, A.A. Lazarev, A. Dolgui, F. Werner, A Graphical Approach to Solve an Investment Optimization Problem, Preprint 15/13, FMA, Otto-von-Guericke-Universitat Magdeburg, 27 pages.

28. E.R. Gafarov, A. Dolgui, F. Werner (2012), Dynamic Programming Approach to Design FPTAS for Single Machine Scheduling Problems, Report n° RR-12-02, LIMOS, CNRS UMR 6158, March 3rd, 26 pages.