

УДК 519.8

ДВУСТАДИЙНАЯ СИСТЕМА ПЛАНИРОВАНИЯ РАБОЧЕГО ВРЕМЕНИ

Н.М. Матвейчук², к.ф.-м.н., Ю.Н. Сотсков¹, д.ф.-м.н., профессор,
А.А. Косенков¹, аспирант

¹ОИПИ НАН Беларуси, ²УО «Белорусский государственный аграрный
технический университет», г. Минск, Республика Беларусь

Современному работнику необходимо не только много работать, но и много успевать, а это часто не одно и то же. Повышение производительности труда работника зависит от того, насколько эффективно используется его рабочее время. Для улучшения организации деятельности работника используются различные методики тайм-менеджмента [1] – отрасли менеджмента, призванной рационально и эффективно использовать временной ресурс работника.

Наиболее значимым является планирование рабочего времени руководителя, поскольку современный руководитель должен не только своевременно решать тактические вопросы и определять стратегию своего предприятия, но и обеспечивать полезную загрузку подчиненных ему сотрудников, контролировать их деятельность, нести ответственность за результаты работы всего коллектива. Рабочее время руководителя имеет более высокую стоимость. Однако планирование рабочего времени руководителя нельзя рассматривать в отрыве от планирования рабочего времени его подчиненного – руководитель и подчиненный связаны выполнением общих работ, и поэтому расписания их работ должны быть взаимосвязаны.

В докладе рассматривается система планирования рабочего времени двух исполнителей, а именно: *главного* исполнителя A (руководителя) и *подчиненного* ему исполнителя B . Имеется список работ $J = \{J_1, J_2, \dots, J_n\}$, предназначенных к выполнению в течение рабочего дня (недели или другого периода времени), среди которых можно выделить три группы работ: работы, которые выполняет только руководитель; работы, которые выполняет только подчиненный; и работы, которые выполняет вначале руководитель, а затем подчиненный (т.е. руководитель ставит задачу, намечает пути ее решения, определяет этапы выполнения задания, которые затем реализует его подчиненный).

Не нарушая общности, предположим, что каждая работа $J_i \in J$ состоит из двух операций O_{iA} и O_{iB} , первая из которых выполняется исполнителем A , а вторая – исполнителем B . Такая обслуживающая система в теории расписаний [2] называется *поточной*, и для двух *приборов* (исполнителей) обозначается $F2$. В нашем случае работы первой группы будут иметь нуле-

вую длительность выполнения второй операции, а работы второй группы – нулевую длительность выполнения первой операции. Для простоты будем предполагать, что прерывания при выполнении операций запрещены для обоих исполнителей.

Для каждой работы $J_i \in J$ планируемая длительность ее выполнения $p_{ij} \geq 0$ на этапе составления расписания может быть известна лишь с определенной погрешностью. Поэтому, вместо *детерминированных* длительностей операций p_{ij} следует рассматривать *неопределенные* (интервальные) длительности операций, предполагая заданными на этапе составления расписания их верхние границы a_{ij} и нижние границы b_{ij} : $a_{ij} \leq p_{ij} \leq b_{ij}$. Возможные длительности работ определяются в виде многогранника $T = \{p \mid a_i \leq p_i \leq b_i, J_i \in J, j \in \{A, B\}\}$ допустимых $2n$ -мерных векторов $p = (p_{1A}, p_{1B}, p_{2A}, \dots, p_{nB}) \in T$.

Критерием оптимальности расписания является минимизация общего (максимального) времени $C_{max} = \max\{C_i \mid J_i \in J\}$ выполнения заданного множества работ, где C_i – момент завершения выполнения работы $J_i \in J$. Этот критерий является регулярным (целевая функция не убывает ни по одному из своих аргументов), поэтому оптимальное расписание достаточно искать среди активных расписаний, т. е. расписаний без неоправданных простоев каждого исполнителя. Активное расписание задается перестановкой работ $\pi_i = (J_{i1}, J_{i2}, \dots, J_{in}) \in S$, определяющей порядок их выполнения. Здесь $\{i_1, i_2, \dots, i_n\} = \{1, 2, \dots, n\}$ и $S = \{\pi_1, \pi_2, \dots, \pi_{n!}\}$ – множество $n!$ перестановок n работ множества J . В обозначениях, принятых в теории расписаний [2], рассматриваемая *неопределенная* задача планирования рабочего времени руководителя и его подчиненного может быть представлена следующим образом: $F2/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$. Эта задача исследовалась в статьях [3-5]. Отметим, что в обозначениях, принятых в [5], работы первой группы относятся к множеству J^2 , работы второй группы относятся к множеству J^1 , работы третьей группы могут быть отнесены к любому из множеств J^1, J^2, J^* , в зависимости от взаимного расположения интервалов выполнения операций.

Для каждого вектора $p \in T$ длительностей операций (для каждого сценария) *неопределенная* задача $F2/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$ превращается в детерминированную задачу $F2/C_{max}$. Следует отметить, что в силу неопределенности длительностей операций в общем случае не существует перестановки, оптимальной для задачи $F2/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$ при традиционном подходе к решению задачи (оптимальной для всех векторов длительностей операций $p \in T$). Задача $F2/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$ с математической точки зрения некорректна. Поэтому в качестве решения такой задачи предлагается рассматривать минимальное по включению множество перестановок, содержащую

хотя бы одну перестановку Джонсона для каждого вектора $p \in T$ в соответствии со следующим определением.

Определение. Решением задачи $F2/a_{ij} \leq p_{ij} \leq b_{ij}/C_{max}$ будем называть минимальное по включению подмножество $S(T)$ множества перестановок S , содержащее для каждого вектора $p \in T$ длительностей операций хотя бы одну перестановку Джонсона для детерминированной задачи $F2/C_{max}$ с вектором p длительностей операций.

Алгоритмы построения множества $S(T)$ и его свойства рассмотрены в работах [4-5].

Опишем, как следует реализовать систему планирования рабочего времени на персональном компьютере. Основная архитектурная особенность разработки приложения для построения расписания и его мониторинга, основанного на модели «руководитель-подчиненные», заключается в необходимости разработки единого механизма взаимодействия всех пользователей, участвующих в процессе выполнения работ. Это означает, что каждый из пользователей должен иметь доступ к общим данным, иметь возможность изменять их в той мере, в какой это необходимо, и иметь возможность информировать об этом других пользователей. Приложением, наиболее подходящим под такой сценарий, является распределенное приложение, типа «клиент-сервер». В рамках данного подхода, приложение будет представлено серверной частью, на которой будут производиться основные расчеты, и клиентской частью, которая может быть представлена различными типами «клиентов», и использующей данные, поставляемые серверной частью.

Серверная часть, как наиболее важная часть приложения, состоит из нескольких компонент, а именно:

- сервер базы данных, которая будет хранить и обрабатывать все данные, используемые в приложении;
- процессор приложения, представленный одной или несколькими исполняемыми библиотеками, которые служат для выполнения всех подсчетов предметной области (построение расписания, мониторинг расписания, корректировка расписания, статистические подсчеты);
- сервисный слой приложения, представленный одной или несколькими исполняемыми библиотеками, которые будут содержать набор функциональных интерфейсов, посредством которых клиентская часть приложения будет взаимодействовать с сервером;
- панель администрирования серверной части, которая является частным случаем «клиента», необходимого для управления сервером и обладающего полным функционалом.

Клиентская часть может быть представлена:

- приложением с графическим интерфейсом, которое может быть разных типов (desktopное, мобильное, веб-приложение);
- локальной базой данных, для хранения необходимых промежуточных данных.

Таким образом, в распределенном приложении достигается гибкость в использовании приложения: можно использовать одну и ту же логику для клиентов разных типов, что особенно актуально сегодня, когда мобильные и облачные технологии становятся все более востребованными.

Основной упрощенный сценарий взаимодействия пользователей данного приложения будет следующим. Все пользователи (сотрудники) зарегистрированы в центральной базе, находящейся на сервере. Руководитель производит построение расписания через административную утилиту, получает и назначает доступных подчиненных на выполнение работ из заданного списка. По завершении выполнения работ подчиненный может отметить работу как выполненную, о чем руководитель получит соответствующую нотификацию. При данном сценарии приложении будет выступать как внутрикорпоративное приложение, которое может использоваться в рамках компании или предприятия.

Поскольку выполнение расписания подразумевает необходимость иметь представление о том, из каких компонент оно состоит, а также подразумевает динамическое изменение во времени, то наиболее важными особенностями графического интерфейса клиента разрабатываемого приложения должны быть информативность и интерактивность. Под информативностью подразумевается разнообразие графических представлений расписания, его компонент, под интерактивностью — наличие гибкой обратной связи с приложением, посредством которой оно могло бы информировать пользователя о тех или иных изменениях и предоставлять выбор выполнения необходимого действия. Построение и реализация расписания должны быть такими, чтобы пользователь не задумывался о технических деталях тайм-менеджмента и в то же время должны позволять ему более рационально использовать его рабочее время.

Следует отметить, что разработка и использование данного приложения не должны рассматриваться как обязательный для исполнения набор инструкций. Его основное назначение предоставить пользователю инструментарий для построения списка его работ в том порядке, который позволил бы оптимизировать его рабочее время.

Работа первого автора выполнена частично при финансовой поддержке Белорусского республиканского фонда фундаментальных исследований.

Литература

1. Новак, Б.В. Тайм-менеджмент на компьютере. Как управлять своим временем эффективно? / Б.В. Новак. – СПб.: Питер, 2007. – 128 с.
2. Graham, R.L. Optimization and approximation in deterministic sequencing and scheduling. A survey / R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G Rinnooy Kan // Annals of Discrete Mathematics. – Vol. 5. – 1979. – P. 287 – 326.
3. Matsveichuk, N.M. Schedule execution for two-machine flow-shop with interval processing times / N.M. Matsveichuk, Yu.N. Sotskov, N.G. Egorova, T.-C. Lai // Mathematical and Computer Modelling. – 2009. – 49. – P. 991-1011.
4. Matsveichuk, N.M. The dominance digraph as a solution to the two-machine flow-shop problem with interval processing times / N.M. Matsveichuk, Y.N. Sotskov, F. Werner // Optimization. – 2011. – 60:12, 1493-1517.
5. Ng, C. T. Two-machine flow-shop minimum-length scheduling with interval processing times / C. T. Ng, N. M. Matsveichuk, Yu. N. Sotskov, T. C. E. Cheng // Asia-Pacific Journal of Operational Research. – 2009. – Vol. 26. – No. 6. – P. 1-20.

УДК 631.15(477.74)

ПРОИЗВОДСТВО ЗЕРНОВЫХ ВО ЛЬВОВСКОЙ ОБЛАСТИ УКРАИНЫ

Л.Н. Михалюк, аспирант

Львовский национальный аграрный университет, Украина

Введение

Зерно для Украины является стратегической рыночной продукцией и одним из основных источников денежных поступлений для сельскохозяйственных товаропроизводителей. Имея в своем арсенале 27% всех черноземов планеты, Украина и сегодня должна занимать лидирующие позиции по производству и поставкам зерна на мировой рынок. Однако, каждый год, плодородие наших земель постепенно уменьшается и мы значительно уступаем мировым лидерам производства зерна по уровню урожайности зерновых. Эффективность зернопроизводства формируется не только под воздействием природных, но и социально-экономических условий отдельных регионов. Отсюда важно, что решение глобальных проблем зернового рынка требует индивидуального подхода к каждому региону отдельно. Подробное изучение состояния производства зерна в каждой области позволит определить и проанализировать истоки основных существующих проблем в этой