

И. А. Каляев, чл.-корр. РАН, kaliaev@mvs.sfedu.ru,

Южный федеральный университет,

А. И. Каляев, канд. техн. наук, anatoly@kalyaev.net,

Я. С. Коровин, канд. техн. наук,

Научно-исследовательский институт многопроцессорных вычислительных систем

Южного федерального университета

Принципы организации и функционирования безлюдного роботизированного производства с децентрализованным диспетчером

Рассматривается проблема создания безлюдных роботизированных производств (БРП), включающих в свой состав некоторое множество роботизированных обрабатывающих центров (РОЦ) различной функциональной направленности. Основу такого БРП должен составлять автоматический диспетчер, функции которого заключаются в построении плана (временного графика) изготовления заказываемого изделия с помощью РОЦ, входящих в состав БРП, а также плана транспортных операций по перемещению комплектующих и заготовок между ними. Организация такого БРП требует решения двух основополагающих задач: во-первых, необходимо разработать "форму" представления задания на изготовление заказываемого изделия, "понятную" для диспетчера БРП, а во-вторых, разработать метод автоматического формирования плана (временного графика) выполнения задания с помощью РОЦ, входящих в состав БРП. Предлагается метод представления задания на изготовление изделия в виде ациклического графа, каждой вершине которого приписана некоторая операция, выполняемая одним из РОЦ. При этом дуги графа определяют операции по транспортировке комплектующих и заготовок между РОЦ. Также предлагается метод диспетчеризации БРП с помощью множества программных агентов, физически реализуемых на контроллерах отдельных РОЦ, причем распределение операций задания между РОЦ осуществляется путем их коллективного взаимодействия посредством некоторого информационного канала связи. Разрабатывается подробный алгоритм работы программного агента РОЦ.

Ключевые слова: безлюдное роботизированное производство, автоматический диспетчер, граф задания, программный агент, коллективное взаимодействие, алгоритм работы агента

Введение

Ускорение научно-технического прогресса приводит к усилению конкурентной борьбы на рынке высокотехнологичной продукции. Побеждает тот, кто сможет быстрее довести свою идею до демонстрационного или опытного образца, способного заинтересовать рыночных инвесторов. В этом плане все большую актуальность приобретает проблема создания безлюдных роботизированных производств (БРП), способных в кратчайшие сроки изготавливать разнообразные опытные образцы и единичные изделия по требованиям заказчика [1, 2]. Такое БРП должно включать в свой состав широкий набор роботизированных обрабатывающих центров (РОЦ) различной функциональной направленности. Заказчик формирует и направляет по сети Интернет задания на изготовление своего изделия с помощью РОЦ, входящих в состав БРП. На основании этого задания автоматический диспетчер БРП должен строить план (временной график) изготовления данного изделия, т.е. распределить операции по отдельным РОЦ, а также организовать транспортировку и доставку требуемых комплек-

тующих и заготовок со склада и между отдельными РОЦ. Далее изделие запускается в производство согласно построенному плану.

Организация такого БРП требует решения двух основополагающих проблем: во-первых, необходимо разработать "форму" представления задания на изготовление заказываемого изделия, "понятную" для диспетчера БРП, а во-вторых, разработать метод автоматического формирования плана (временного графика) выполнения задания с помощью РОЦ, входящих в состав БРП. Именно решению этих проблем посвящена настоящая статья.

Формальная постановка задачи

Будем считать, что в состав БРП входит некоторое множество РОЦ R_1, R_2, \dots, R_N , а также два склада — склад комплектующих и заготовок и склад готовых изделий (рис. 1). Все РОЦ и склады объединены общей транспортной линией, посредством которой комплектующие, заготовки и изделия могут передаваться между складами и РОЦ, а также между отдельными РОЦ (рис. 1).

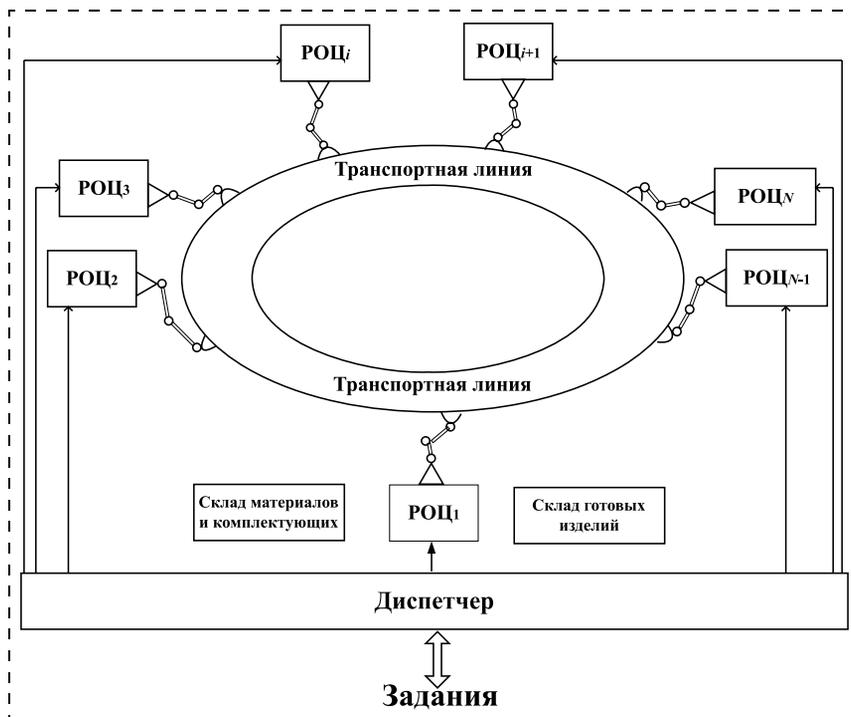


Рис. 1. Структура БРП

Предположим, что каждый РОЦ R_i может выполнять некоторый набор операций $A_i = \langle A_1^i, A_2^i, \dots, A_L^i \rangle$ ($i = 1, 2, \dots, N$), причем в общем случае $A_i \neq A_j$ ($j = 1, 2, \dots, i - 1, i + 1, \dots, N$). Будем считать, что РОЦ R_i выполняет операцию A_l^i ($l = 1, 2, \dots, L$) за время $t_i(A_l^i)$, причем время выполнения идентичных операций различными РОЦ одинаково. Кроме того положим, что время транспортировки комплектующих и заготовок между отдельными РОЦ, а также между РОЦ и складами составляет $t_{\pi}(S)$, где S — длина транспортной линии между ними.

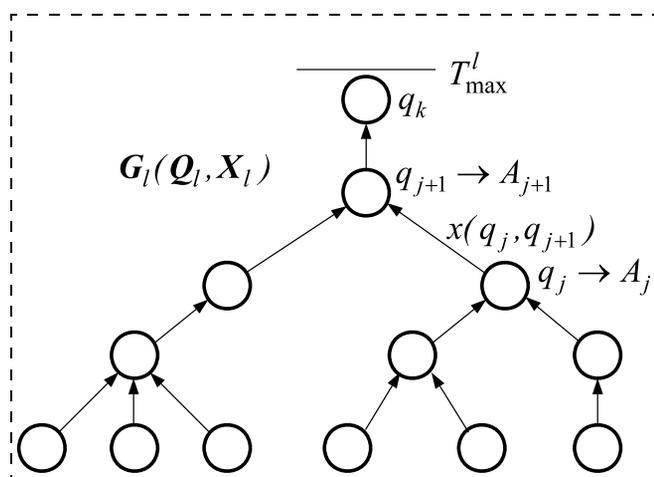


Рис. 2. Граф $G_l(Q_l, X_l)$ задания Z_l

Будем считать, что в БРП в случайные моменты времени через сеть Интернет от различных заказчиков поступает некоторое множество (поток) различных заданий на изготовление изделий $Z = \langle Z_1, Z_2, \dots, Z_M \rangle$, причем для каждого задания Z_l заказчиком устанавливается момент времени T_{\max}^l , к которому он желает получить свое изделие.

Цель работы БРП заключается в выполнении всех заданий на изготовление изделий $Z = \langle Z_1, Z_2, \dots, Z_M \rangle$ к установленным заказчиками моментам времени.

Формализация задания на изготовления изделия

Для того чтобы БРП мог осуществить изготовление некоторого изделия $Z_l \in Z$, задание на его изготовление должно быть представлено заказчиком в некотором унифицированном формализованном виде, понятном автоматическому диспетчеру БРП.

Задание на изготовление некоторого изделия $Z_l \in Z$ можно представить в виде ациклического графа $G_l(Q_l, X_l)$ (рис. 2), каждой вершине $q_j \in Q_l$ которого приписана некоторая операция A_j , принадлежащая множеству операций $A = \bigcup_{i=1}^N A_i$, выполняемых различными РОЦ, причем если две вершины графа q_j и q_{j+1} соединены дугой $x(q_j, q_{j+1})$, то это означает, что операция A_{j+1} , приписанная вершине q_{j+1} , должна выполняться по завершению операции A_j , приписанной вершине q_j . Входные вершины графа $G_l(Q_l, X_l)$ определяют операции по доставке исходных комплектующих и заготовок, необходимых для изготовления изделия, со склада, а конечная вершина определяет операцию по размещению на складе готовой продукции конечного изделия, получаемого в результате выполнения всей программы его изготовления.

Принципы организации диспетчера БРП

После того как задание Z_l на изготовление изделия формализовано в виде графа $G_l(Q_l, X_l)$, оно передается через сеть Интернет в диспетчер БРП, функции которого заключаются в построении плана (временного графика) изготовления изделия, т.е. в распределении отдельных операций задания Z_l между РОЦ с привязкой их выполнения к определенным моментам времени в соответствии с графом задания $G_l(Q_l, X_l)$.

Можно предложить два способа организации такого диспетчера.

В более простом варианте управление БРП может осуществляться с помощью некоторого специально выделенного серверного узла, функции которого заключаются в распределении операций потока поступающих заданий $Z = \langle Z_1, Z_2, \dots, Z_M \rangle$ между РОЦ (см. рис. 1). Однако такая централизованная организация диспетчера БРП имеет целый ряд недостатков. Во-первых, при большом числе РОЦ решение задачи диспетчирования их работы с помощью одного серверного узла будет затруднено вследствие необходимости выполнения большого объема вычислений в реальном времени поступления заданий. Во-вторых, существенно затрудняется масштабирование БРП (т.е. добавление в ее состав новых РОЦ), поскольку при этом необходимо полностью менять не только программу работы центрального диспетчера, но и архитектуру связей диспетчера с РОЦ. И наконец, в-третьих, БРП с центральным диспетчером становится мало надежным, поскольку выход диспетчера из строя приводит к катастрофическим последствиям для всего БРП в целом. Все перечисленные проблемы многократно усложняются в случае, когда БРП должно выполнять не одиночное задание, а поток заданий, поступающих в заранее неизвестные моменты времени.

Всех этих недостатков можно избежать, если использовать принципы децентрализованного мультиагентного диспетчирования в распределенных системах [3–6]. При этом каждый из РОЦ, входящих в состав БРП, должен обладать своим программным агентом, представляющим "интересы" данного РОЦ в процессе диспетчирования, а оптимизация распределения операций поступающих заданий между РОЦ должна осуществляться путем их коллективного взаимодействия посредством некоторого информационного канала связи (рис. 3).

При этом возникает вопрос: каким образом агенты должны взаимодействовать с заказчиками и получать от них задания? Такое взаимодействие может осуществляться посредством некоторого специально пассивного узла, подключенного к сети Интернет и играющего роль "доски объявлений" (ДО), на которой заказчики могут размещать свои задания (рис. 3). При этом дескриптор задания Z_i , размещаемого на ДО, должен содержать:

- граф $G_i(Q_i, X_i)$ задания;
- список вершин множества Q_i и приписанных им операций;
- момент времени T_{\max}^i , к которому потребитель желает получить готовое изделие.

Агенты должны периодически опрашивать ДО с целью загрузить "свой" РОЦ работой. При этом если задание $Z_i \in Z$ не может быть выпол-

нено с помощью одного РОЦ R_i , то агенты должны сформировать некоторую виртуальную организацию — сообщество, состоящее из множества РОЦ $R_i = \langle R_i, R_j, \dots, R_k \rangle$, общей целью которых является выполнение задания $Z_i \in Z$ к установленному заказчиком моменту времени.

В укрупненном виде работу БРП с мультиагентным диспетчером можно представить в следующем виде.

1. Заказчик формирует свое задание $Z_i \in Z$ в виде графа $G_i(Q_i, X_i)$, устанавливает момент времени T_{\max}^i , к которому он хочет получить готовое изделие, и размещает дескриптор задания на ДО.

2. Агент РОЦ R_i , не задействованного в выполнении других заданий, опрашивает ДО в поисках работы для "своего" РОЦ. В случае обнаружения на ДО задания $Z_i \in Z$ агент делает попытку войти в состав сообщества R_i по его выполнению. Для этого агент R_i выделяет фрагмент задания $Z_i \in Z$ (т.е. некоторую последовательность входящих в него операций), не закрепленных ранее за другими агентами, который он может выполнить к установленному моменту времени с помощью "своего" РОЦ. Если такой фрагмент обнаруживается, агент R_i вступает в сообщество R_i и переходит к исполнению принятых на себя операций.

4. Агент R_i контролирует ход выполнения операций принятого к исполнению фрагмента задачи, периодически оценивая время их завершения. В случае если агент по каким-либо причинам не успевает завершить исполнение данных операций к установленному моменту времени, он сообщает об

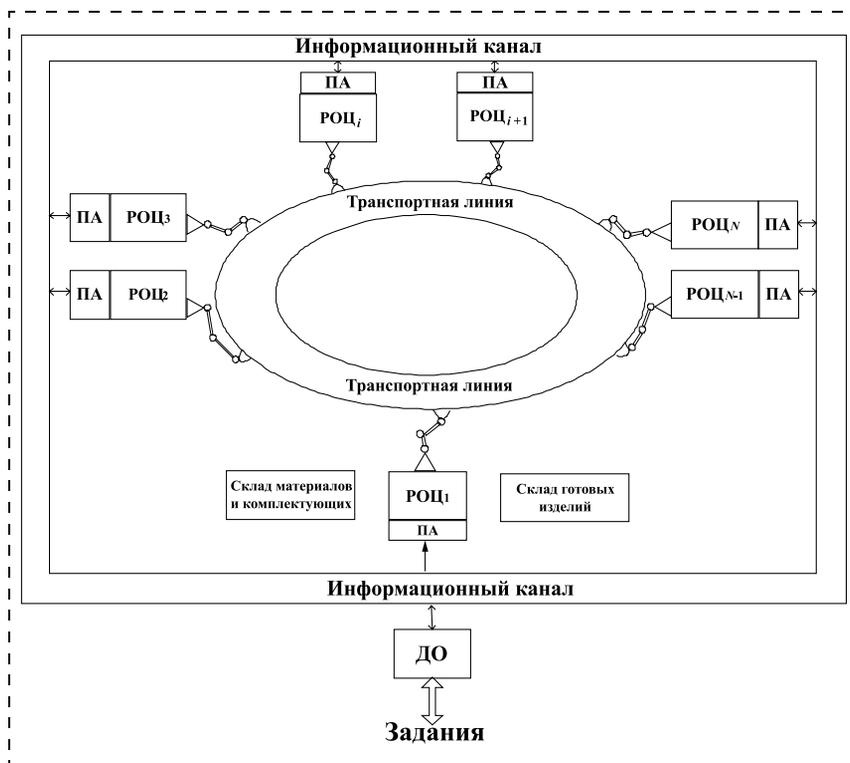


Рис. 3. Структура БРП с мультиагентным диспетчером

этом на доску объявлений и выходит из состава общества R_l .

5. В случае успешного исполнения всех принятых операций задания Z_l агент R_j передает с помощью транспортной линии изделие на другой РОЦ для выполнения последующих операций либо на склад готовой продукции.

Использование принципов децентрализованного мультиагентного диспетчирования БРП обеспечивает: во-первых, высокую отказоустойчивость системы, поскольку в ней отсутствует "узкое горло" в виде центрального диспетчера, а отказ любого из агентов не приводит к катастрофическим последствиям для всего БРП в целом; во-вторых, возможность практически неограниченного увеличения (масштабирования) числа РОЦ в составе БРП путем их простого подключения к информационному каналу связи; и наконец, в-третьих, снижение вычислительной нагрузки на отдельного программного агента при решении задачи диспетчирования, что, в свою очередь, обеспечивает возможность ее решения в реальном времени поступления заданий.

Однако, вместе с тем, использование принципов мультиагентного диспетчирования при управлении БРП требует разработки специального алгоритма работы программного агента РОЦ. Именно разработке такого алгоритма посвящена последующая часть данной статьи.

Алгоритм работы программного агента

Прежде чем приступить к разработке алгоритма мультиагентного диспетчирования БРП, введем понятие "нити". Под нитью будем понимать некоторую последовательность вершин $H_f = \langle q_1^f, q_2^f, \dots, q_k^f \rangle$ графа $G(Q, X)$ задания $Z_l \in Z$, в которой вершины q_j^f и q_{j+1}^f ($j = 1, 2, \dots, k-1$) соединены дугой

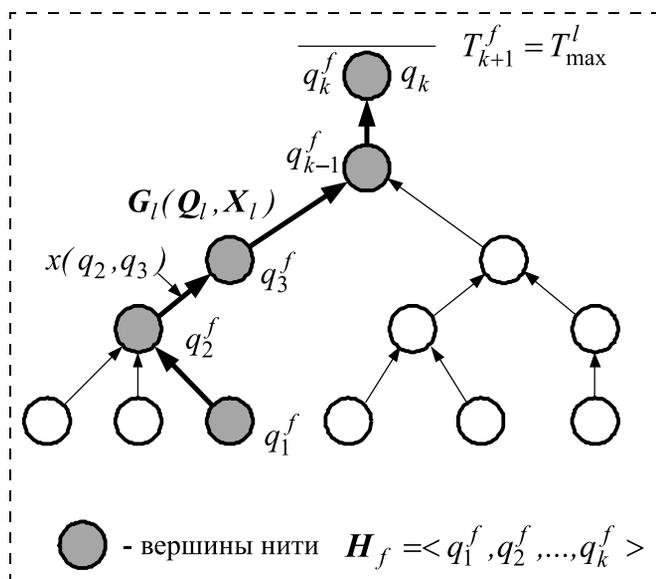


Рис. 4. Выделение нити H_f в графе $G(Q, X)$ задания Z_l

$x(q_j^f, q_{j+1}^f)$ (рис. 4). Иными словами, нить определяет некоторый набор операций задания Z_l , которые должны выполняться последовательно. При этом под длиной t_f нити H_f будем понимать суммарное время, затрачиваемое на выполнение приписанных ее вершинам операций, определяемое как

$$t_f = \sum_{i=1}^k (t_p(A_i^f) + t_{\Pi}(S_p, c)),$$

где $t_p(A_i^f)$ — время, затрачиваемое РОЦ $R_p \in R$ на выполнение операции A_i^f , приписанной вершине $q_i^f \in H_f$ ($i = 1, 2, \dots, k$); $t_{\Pi}(S_p, c)$ — время, затрачиваемое на транспортировку изделия от РОЦ $R_p \in R$, выполняющего операцию A_i^f , к РОЦ $R_c \in R$, выполняющему следующую по очереди операцию A_{i+1}^f нити H_f ; S_p, c — длина транспортной линии между РОЦ R_p и РОЦ R_c .

Если операции A_i^f и A_{i+1}^f выполняются одним и тем же РОЦ R_p , то соответственно $t_{\Pi}(S_p, p) = 0$.

Очевидно, если вся нить H_f выполняется одним ресурсом R_p^f , то ее длина будет составлять

$$t_f = \sum_{i=1}^k t_p(A_i^f). \quad (1)$$

При этом если известен требуемый момент времени T_{k+1}^f исполнения всей нити H_f (рис. 4), то можно определить допустимые моменты времени T_d^f начала выполнения всех операций A_d^f , приписанных ее вершинам $q_d^f \in H_f$ ($d = 1, 2, \dots, k$) (при которых РОЦ R_p успевает выполнить всю нить H_f к требуемому моменту времени T_{k+1}^f), как

$$T_d^f = T_{k+1}^f - \sum_{i=d}^k t_p(A_i^f) \quad (d = 1, 2, \dots, k-1). \quad (2)$$

Исходя из этих соображений, можно предложить следующую процедуру мультиагентного диспетчирования работы БРП при выполнении потока заданий.

Пользователь формирует свое задание $Z_l \in Z$ в виде графа $G(Q, X)$ и определяет требуемый момент времени T_{\max}^l , к которому ее решение должно быть получено. Дескриптор представленного таким образом задания $Z_l \in Z$ размещается на доске объявлений (см. рис. 2).

Агенты, РОЦ которых не задействованы в выполнении каких-либо заданий, обращаются к ДО в поисках работы. Если агент свободного РОЦ $R_p \in R$ обнаруживает на ДО дескриптор задания Z_b , то он делает попытку войти в сообщество R_l по его исполнению.

Поскольку, как мы приняли выше, каждый РОЦ имеет некоторую специализацию (т.е. может выполнять некоторый ограниченный набор операций), то в общем случае может оказаться, что РОЦ R_p способен выполнять далеко не все операции, приписанные вершинам графа $G_l(Q_b, X_l)$ задания Z_l . Поэтому в графе $G_l(Q_b, X_l)$ необходимо предварительно выделить подграф $G_l^p(Q_b^p, X_l^p)$, вершинам которого приписаны операции множества A_p , выполняемые РОЦ R_p (рис. 5). После этого необходимо проанализировать, есть ли в графе $G_l^p(Q_b^p, X_l^p)$ вершины, для которых установлено требуемое время их исполнения. Отметим, что изначально в момент размещения дескриптора задания Z_l на ДО требуемое время исполнения T_{\max}^l приписано только конечной вершине q_k графа $G_l(Q_b, X_l)$ (см. рис. 2). Если таких вершин нет, то это говорит о том, что агент R_p пока что не может вступить в сообщество R_l по выполнению задания Z_b , и поэтому он вновь переходит к режиму опроса ДО в целях поиска других заданий.

В противном случае агент R_p выделяет в графе $G_l(Q_b, X_l)$, хранящемся в дескрипторе задания Z_l на ДО, наиболее длинную нить $H_1 = \langle q_1^1, q_2^1, \dots, q_k^1 \rangle$ согласно выражению (1), конечной вершине q_k^1 которой приписан требуемый момент времени ее исполнения T_{k+1}^1 (рис. 5). Последнее может быть осуществлено с помощью одного из известных алгоритмов поиска экстремальных путей на графах [7].

Далее агент R_p определяет согласно (2) момент времени T_1^1 , когда ему необходимо приступить к выполнению первой операции нити H_1 , т.е. операции A_1^1 , приписанной вершине $q_1^1 \in H_1$, чтобы успеть завершить исполнение всей нити H_1 к заданному моменту времени T_{k+1}^1 .

Если при этом оказывается, что $T_1^1 < T_{\text{тек}}$, где $T_{\text{тек}}$ — текущий момент времени, то это означает, что РОЦ R_p не может обеспечить выполнение всей последовательности операций нити $H_1 = \langle q_1^1, q_2^1, \dots, q_k^1 \rangle$ к установленному заказчиком моменту времени T_{k+1}^1 . Поскольку мы приняли, что время выполнения однотипных операций различными РОЦ одинаково, то никакой другой РОЦ также не сможет выполнить данную нить к требуемому моменту

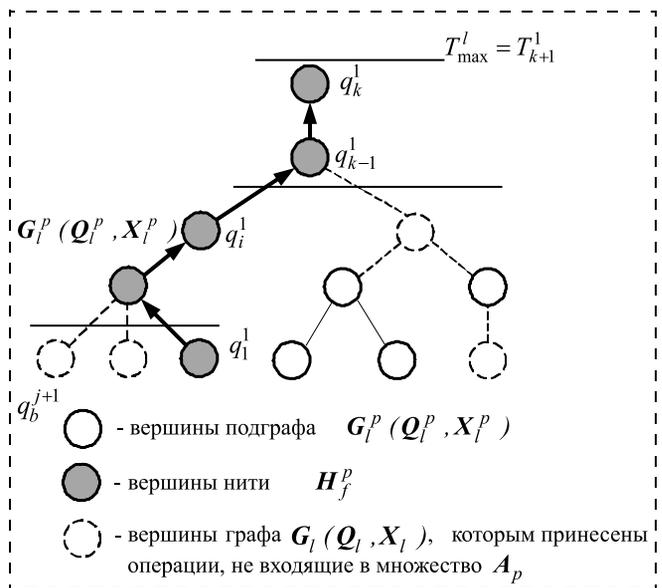


Рис. 5. Выделение подграфа $G_l^p(Q_b^p, X_l^p)$ в графе $G_l(Q_b, X_l)$ задания Z_l

времени. Это говорит о том, что задание Z_l не может быть выполнено к установленному заказчиком моменту времени. В этом случае агент R_p направляет на ДО соответствующее сообщение. Задание Z_l снимается с ДО, а потребителю посылается сообщение о невозможности выполнения его задания к установленному моменту времени. После этого агент РОЦ R_p вновь переходит к опросу ДО в поисках работы.

Если же условие $T_1^1 \geq T_{\text{тек}}$ выполняется для нити H_1 , то в этом случае агент R_p принимает на себя исполнение последовательности операций, приписанных ее вершинам. При этом агент R_p осуществляет модификацию дескриптора задания Z_l на ДО, а именно:

- 1) его идентификатор записывается в список членов сообщества R_l по выполнению задания Z_b ;
- 2) вершины, входящие в нить H_1 , исключаются из графа $G_l(Q_b, X_l)$ задания Z_l , в результате чего формируется новый граф $G_l^1(Q_b^1, X_l^1) = G_l(Q_b, X_l) / H_1$ (рис. 6);
- 3) всем вершинам графа $G_l^1(Q_b^1, X_l^1)$, инцидентным вершинам нити H_1 , приписываются требуемые моменты времени их исполнения, которые определяются исходя из следующих соображений.

Допустим, что некоторая вершина q_f^2 графа $G_l^1(Q_b^1, X_l^1)$ инцидентна вершине q_b^1 , принадлежащей нити H_1 (рис. 6). Это означает, что выполняемая операция, приписанная вершине q_b^1 , может быть начата только по завершении операции, при-

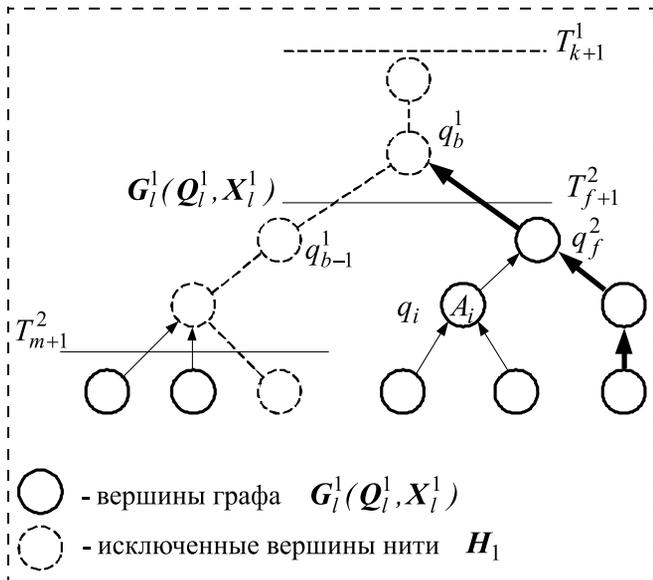


Рис. 6. Граф $G_l^1(Q_l^1, X_l^1)$ задания Z_l , модифицированный агентом R_p

писанной вершине q_f^2 графа $G_l^1(Q_l^1, X_l^1)$. Поэтому, очевидно, что результаты выполнения операции, приписанной вершине q_f^2 , должны быть получены и переданы РОЦ R_p , выполняющему операции нити H_1 , не позже, чем к требуемому моменту времени T_b^1 начала выполнения РОЦ R_p операции A_b^1 , приписанной вершине q_b^1 , определяемому согласно выражению (2) как

$$T_b^1 = T_{k+1}^1 - \sum_{i=b}^k t_p(A_i^1). \quad (3)$$

В противном случае РОЦ R_p не успеет закончить исполнение взятой на себя нити H_1 к требуемому моменту времени T_{k+1}^1 .

Поэтому вершине q_f^2 графа $G_l^1(Q_l^1, X_l^1)$ приписывается требуемое время ее исполнения $T_{f+1}^2 = T_b^1$, а также идентификатор РОЦ R_p , которому результаты исполнения операции, приписанной вершине q_f^2 , должны быть переданы (рис. 6).

Аналогичным образом определяются требуемые моменты T_{m+1}^2 исполнения всех остальных вершин графа $G_l^1(Q_l^1, X_l^1)$, инцидентных вершинам нити H_1 (рис. 6).

Если после модификации новый граф $G_l^1(Q_l^1, X_l^1)$ задания Z_l на ДО еще не пустой, т.е. $G_l^1(Q_l^1, X_l^1) \neq \emptyset$, то процесс создания сообщества R_l для выполнения задания Z_l продолжается далее.

Допустим, что через какое-то время другой свободный РОЦ R_c обнаруживает на ДО дескриптор задания Z_l и предпринимает попытку войти в состав сообщества R_l по его исполнению.

Для этого агент R_c выделяет в графе $G_l^1(Q_l^1, X_l^1)$ подграф $G_l^{1c}(Q_l^{1c}, X_l^{1c})$, вершинам которого приписаны операции, входящие в исполняемое РОЦ R_c множество A_c (рис. 7). Далее в графе $G_l^{1c}(Q_l^{1c}, X_l^{1c})$ агент R_c выделяет наиболее длинную нить $H_2 = \langle q_1^2, q_2^2, \dots, q_f^2 \rangle$, конечной вершине q_f^2 которой приписано требуемое время исполнения T_{f+1}^2 (рис. 7), и анализирует возможность ее исполнения с помощью "его" РОЦ к данному моменту времени. Для этого он с помощью выражения (2) определяет время T_1^2 начала исполнения операции A_1^2 , приписанной первой вершине q_1^2 данной нити H_2 , и сравнивает его с текущим временем $T_{\text{тек}}$. Если $T_{\text{тек}} > T_1^2$, то это означает, что РОЦ R_c не может выполнить данную нить к требуемому моменту времени T_{f+1}^2 , что говорит о том, что все задание в целом также не может быть выполнено к установленному заказчиком моменту времени. В этом случае задание снимается с ДО, заказчику посылается сообщение о невозможности выполнения его задания к установленному времени, а агент R_c снова переходит в режим опроса ДО в поисках работы для "своего" ресурса.

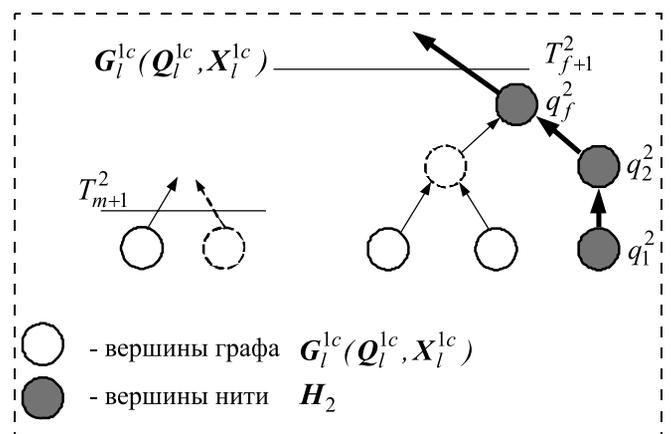


Рис. 7. Выделение подграфа $G_l^{1c}(Q_l^{1c}, X_l^{1c})$ в графе $G_l^1(Q_l^1, X_l^1)$

В случае если условие $T_1^2 \geq T_{\text{тек}}$ выполняется, агент R_c принимает на себя исполнение операций, приписанных вершинам нити H_2 , и осуществляет очередную модификацию дескриптора задания Z_l на ДО:

- идентификатор агента R_c записывается в список членов сообщества R_l по решению задачи Z_l ;
- вершины, входящие в нить H_2 , исключаются из графа $G_l^1(Q_l^1, X_l^1)$, в результате чего образуется новый граф $G_l^2(Q_l^2, X_l^2)$ (рис. 8);
- вершинам q_p^3 графа $G_l^2(Q_l^2, X_l^2)$, инцидентным вершинам q_d^2 нити H_2 , приписывается идентификатор РОЦ R_c , которому результаты исполнения этих операций должны быть направлены, а также требуемое время их исполнения (рис. 8), определяемое как

$$T_{p+1}^3 = T_{f+1}^2 - \sum_{i=d}^f t_c(A_i^2).$$

Далее в процесс распределения операций задания Z_l включается следующий свободный агент, обнаруживший ее дескриптор на ДО, и т.д. до тех пор, пока не окажется, что после очередной модификации граф $G_l^j(Q_l^j, X_l^j)$ стал пустым, что означает, что все операции задания Z_l разобраны агентами, вошедшими в сообщество R_l по его выполнению.

После того как некоторый агент R_p выбрал для исполнения нить $H_f = \langle q_1^f, q_2^f, \dots, q_k^f \rangle$, он приступает к исполнению операций, приписанных ее вершинам с помощью "своего" РОЦ. При этом перед началом выполнения очередной операции A_d^f , приписанной вершине $q_d^f \in H_f$ ($d = 1, 2, \dots, k$), агент R_p должен проверить, во-первых, наличие всех комплектующих и заготовок, необходимых для ее выполнения, а во-вторых, соблюдение временного графика выполнения всей нити H_f в целом.

Поскольку для выполнения подзадачи A_d^f могут потребоваться заготовки, получаемые в результате выполнения смежной нити другим РОЦ R_c , то к моменту начала выполнения РОЦ R_p операции A_d^f может оказаться, что эти заготовки еще не поступили. В этом случае агент R_p должен перейти в режим ожидания поступления необходимых заготовок. Это ожидание может продолжаться до тех пор, пока выполняется условие $T_d^f \geq T_{\text{тек}}$, где T_d^f — требуемое время начала исполнения операции A_d^f ,

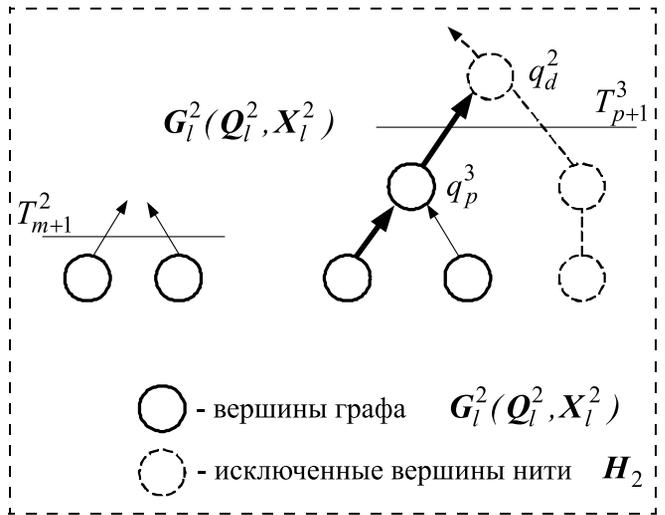


Рис. 8. Граф $G_l^2(Q_l^2, X_l^2)$ модифицированный агентом R_c

приписанной вершине $q_d^f \in H_f$ и определяемое согласно выражению (2).

Если оказывается, что $T_d^f < T_{\text{тек}}$, то это означает, что РОЦ R_p уже не успевает выполнить оставшиеся операции нити H_f к требуемому моменту времени T_{k+1}^f . В этом случае агент R_p должен известить ДО о том, что произошло отставание от графика выполнения задания, и поэтому задание Z_l не может быть решено к установленному заказчиком моменту времени. При этом задание снимается с ДО, пользователю сообщается о невозможности его исполнения к установленному им моменту времени, а всем агентам, вошедшим в сообщество R_l по его решению направляется сообщение о прекращении процесса выполнения задания Z_l , после чего они переходят в режим поиска новых заданий на ДО.

После успешного выполнения всех операций нити H_f агент R_p вновь переходит в режим опроса ДО в целях вхождения в новое сообщество по выполнению следующего задания.

Процесс выполнения задания Z_l продолжается до тех пор, пока не окажется, что список агентов-участников сообщества R_l по его выполнению пуст,

а также пуст граф $G_l^j(Q_l^j, X_l^j)$, хранимый в дескрипторе задания Z_l на ДО. Это означает, что все нити задания успешно выполнены. После этого задание снимается с ДО, а заказчику отправляется сообщение об успешном выполнении его задания.

Описанному выше процессу отвечает следующий укрупненный алгоритм функционирования агента, представляющего РОЦ R_p в процессе диспетчирования работы БРП.

Алгоритм 1

1. Агент свободного РОЦ R_p опрашивает ДО.
 2. При обнаружении на ДО задания Z_l агент R_p анализирует граф задания $G_l^j(Q_l^j, X_l^j)$. Если $G_l^j(Q_l^j, X_l^j) = \emptyset$, то переход к п. 1, иначе

3. В графе $G_l^j(Q_l^j, X_l^j)$ агент R_p выделяется подграф $G_l^{jp}(Q_l^{jp}, X_l^{jp})$, вершинам которого приписаны операции множества A_p , выполняемые РОЦ R_p .

4. Если $G_l^{jp}(Q_l^{jp}, X_l^{jp}) = \emptyset$, то перейти к п. 1, иначе

5. Агент R_p выделяет в графе $G_l^{jp}(Q_l^{jp}, X_l^{jp})$ наиболее длинную нить $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$, конечной вершине которой приписано требуемое время исполнения T_{k+1}^{jp} (в момент размещения задания Z_l на ДО требуемое время $T_{k+1}^{jp} = T_{\max}^l$ приписано только конечной вершине q_k графа $G_l(Q_l, X_l)$). Если таковой нити в графе $G_l^{jp}(Q_l^{jp}, X_l^{jp})$ нет, то перейти к 1, иначе

6. Агент R_p определяет допустимый момент времени, когда необходимо начать выполнение нити $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$, чтобы успеть завершить ее исполнение к требуемому моменту T_{k+1}^{jp} , как

$$T_1^{jp} = T_{k+1}^{jp} - \sum_{i=1}^k t_p(A_i^{jp}).$$

7. Если $T_1^{jp} < T_{\text{тек}}$, где $T_{\text{тек}}$ — текущий момент времени, то переход к п. 16, иначе

8. Агент R_p принимает на себя исполнение нити $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$, для чего модифицирует дескриптор задания Z_l на ДО: записывает в список участников сообщества R_l свой идентификатор; модифицирует граф $G_l^j(Q_l^j, X_l^j)$ задания Z_l путем исключения из него вершин нити $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$, т.е. $G_l^{j+1}(Q_l^{j+1}, X_l^{j+1}) = G_l^j(Q_l^j, X_l^j) / H_j^p$; приписывает вершинам $q_j^{j+1,p}$ модифицированного графа $G_l^{j+1}(Q_l^{j+1}, X_l^{j+1})$, инцидентным вершинам $q_b^{j,p}$ нити $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$, идентификатор РОЦ R_p , которому необходимо передать результаты исполнения операции $A_f^{j+1,p}$, приписанной вершине

$q_f^{j+1,p}$, а также требуемое время их исполнения, определяемое согласно выражению (2).

9. Агент R_p переходит к исполнению последовательности операций, приписанных вершинам нити $H_j^p = \langle q_1^{jp}, q_2^{jp}, \dots, q_k^{jp} \rangle$; $d = 1$.

10. Если $T_d^{jp} < T_{\text{тек}}$, где $T_d^{jp} = T_{k+1}^{jp} - \sum_{i=d}^k t_p(A_i^{jp})$ — требуемое время начала выполнения операции A_d^{jp} , приписанной вершине $q_d^{jp} \in H_j^p$, то перейти к п. 16, иначе

11. Агент R_p проверяет наличие заготовок, необходимых для выполнения операции A_d^{jp} . Если заготовки еще не поступили, то перейти к п. 10, иначе

12. Агент R_p выполняет операцию A_d^{jp} , приписанную вершине $q_d^{jp} \in H_j^p$ с помощью своего РОЦ.

13. Если агенту R_p поступило сообщение о прекращении выполнения задания Z_l , то переход к п. 1, иначе

14. $d = d + 1$, если $d \leq k$, то переход к п. 10, иначе

15. Агент R_p сообщает на ДО об успешном выполнении всех операций, приписанных вершинам нити H_j^p задания Z_l . При этом идентификатор агента R_p исключается из списка членов сообщества R_l по выполнению данного задания. Переход к п. 1.

16. Задание Z_l не может быть выполнено к установленному заказчиком моменту времени T_{\max}^l . Дескриптор задания Z_l удаляется с ДО, заказчику направляется сообщение о невозможности выполнения его задания к требуемому моменту времени, а всем агентам, номера которых записаны в списке участников сообщества R_l по выполнению задания Z_l , передается сообщение о прекращении ее исполнения. Переход к п. 1.

Заключение

В статье описаны обобщенные принципы организации и функционирования безлюдного роботизированного производства с децентрализованным диспетчером. Реализация данных принципов обеспечивает:

- возможности автоматического выполнения потока заданий на производство различных изделий, поступающих от заказчиков в заранее неизвестные моменты времени;
- квазиоптимальное адаптивное распределение ресурсов (РОЦ) БРП с учетом их специализации в реальном времени поступления заданий;
- высокую полезную загрузку работой РОЦ, входящих в состав БРП;
- возможность неограниченного наращивания (масштабируемости) числа различных РОЦ в составе БРП;

- повышенную отказоустойчивость БРП, поскольку в ней отсутствуют узлы, выход из строя которых приводит к выходу из строя всего БРП в целом.

Список литературы

1. URL: www.umpro.ru
2. URL: www.rosbalt.ru
3. **Каляев А. И., Каляев И. А.** Метод децентрализованного управления распределенной системой при выполнении потока заданий // Мехатроника, автоматизация, управление. 2015. № 9. С. 585—598.

4. **Каляев А. И., Каляев И. А., Коровин Я. С.** Метод мультиагентного диспетчирования ресурсов в гетерогенной облачной среде при выполнении потока задач // Вестник компьютерных и информационных технологий. 2015. № 11. С. 34—40.
5. **Kalyaev A. I.** Multiagent Approach for Building Distributed Adaptive Computing System // *Procedia Computer Science*. 2013. Vol. 18. P. 2193—2202 (URL: <http://dx.doi.org/10.1016/j.procs.2013.05.390>)
6. **Kalyaev A. I., Korovin Y. S.** Adaptive Multiagent Organization of the Distributed Computations // *AASRI Procedia*. 2014. Vol. 6. P. 49—58 (URL: <http://dx.doi.org/10.1016/j.aasri.2014.05.008>).
7. **Рейнгольд Э., Нивергельт Ю., Део Н.** Комбинаторные алгоритмы. Теория и практика. М.: Мир, 1980. 476 с.

Principles of Organization and Operation of an Unmanned Robotic Production with a Decentralized Controller

I. A. Kalyaev, kaliaev@mvs.sfedu.ru✉, Southern Federal University, Taganrog, 344006, Russian Federation,

A. I. Kalyaev, anatoly@kalyaev.net, Institute of Multiprocessing Computing Systems of Southern Federal University, Taganrog, 347928, Russian Federation,

I. S. Korovin, Institute of Multiprocessing Computing Systems of Southern Federal University, Taganrog, 347928, Russian Federation

Corresponding author: **Kalyaev I. A.**, Corresponding Member of the Russian Academy of Sciences, Head of Research, Southern Federal University, Taganrog, 344006, Russian Federation, e-mail: kaliaev@mvs.sfedu.ru

Received on June 07, 2016

Accepted on June 27, 2016

This work is devoted to the topical problem of elaboration of new principles of a decentralized cooperation of the computer systems, which will create a new class of unmanned robotic productions (URP). In the shortest possible time they could produce a variety of prototypes and individual items meeting the remote customers' requirements. Use of the principles of decentralized multiagent scheduling of URP ensures: firstly, high availability of the system, because it lacks the "bottleneck" in the form of a central controller, and a failure of any of the agents does not lead to the catastrophic consequences for the entire URP; secondly, a possibility of virtually unlimited increase of the number of nodes as a part of URP because of an easy connectivity to the information communication channel; and, finally, thirdly, a reduction of the computational load on an individual software agent in the solution of the scheduling tasks, which in turn, makes it possible to get solutions in real time. This article describes the generic principles of organization and functioning of the unmanned robotic productions with a decentralized controller. Implementation of these principles ensures: — a possibility of automation of the approach to the flow of tasks of production of various products coming from customers in advance and in unknown times; — an adaptive quasi-optimal allocation of resources based on their expertise in the real-time of receipt of the assignments; — a high payload operation of the nodes in the URP; — a possibility of unlimited scalability; — an increased fault tolerance, since it lacks components, the failure of which could lead to a failure of the entire system.

Keywords: *unmanned robotic manufacturing, decentralized controller, fault tolerance, multi-agent system, robotic machining center*

For citation:

Kalyaev I. A., Kalyaev A. I., Korovin I. S. Principles of Organization and Operation of an Unmanned Robotic Production with a Decentralized Controller, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 11, pp. 741—749.

DOI: [10.17587/mau.17.741-749](https://doi.org/10.17587/mau.17.741-749)

References

1. **Available at:** www.umpro.ru
2. **Available at:** www.rosbalt.ru
3. **Kalyaev A. I., Kalyaev I. A.** *Metod decentralizovannogo upravleniya raspredelennoj sistemoj pri vypolnenii potoka zadaniy* (Decentralized management of distributed system method when the flow

zadaniy), *Mekhatronika, avtomatizaciya, upravlenie*, 2015, no. 9, pp. 585—598 (in Russian).

4. **Kalyaev A. I., Kalyaev I. A., Korovin Y. S.** *Metod mul'tiagentnogo dispetchirovaniya resursov v geterogennoj oblachnoj srede pri vypolnenii potoka zadach* (The method of multi-agent scheduling resources in a heterogeneous cloud environment when the task flow), *Vestnik komp'yuternyh i informacionnyh tekhnologij*, 2015, no. 11, pp. 34—40 (in Russian).

5. **Kalyaev A. I.** Multiagent Approach for Building Distributed Adaptive Computing System, *Procedia Computer Science*, 2013, vol. 18, pp. 2193—2202 (URL: <http://dx.doi.org/10.1016/j.procs.2013.05.390>).

6. **Kalyaev A. I., Korovin Y. S.** Adaptive Multiagent Organization of the Distributed Computations, *AASRI Procedia*, 2014, vol. 6, pp. 49—58 (URL: <http://dx.doi.org/10.1016/j.aasri.2014.05.008>).

7. **Reinhold E., Nivergelt Yu., Deo N.** *Kombinatornye algoritmy. Teoriya i praktika* (Combinatorial algorithms. Theory and practice), Moscow, Mir, 1980, 476 p. (in Russian).