http://dx.doi.org/10.35596/1729-7648-2020-18-5-53-61

Оригинальная статья
Original paper

УДК 628.52.012.011

ДИНАМИЧЕСКАЯ АСИММЕТРИЧНАЯ ЗАДАЧА О НАЗНАЧЕНИИ В ОТКРЫТЫХ МНОГОАГЕНТНЫХ СИСТЕМАХ

РЕВОТЮК М.П., ХАДЖИНОВА Н.В., КУЗНЕЦОВ А.П., ШИЛИН Л.Ю.

Белорусский государственный университет информатики и радиоэлектроники
(г. Минск, Республика Беларусь)

Поступила в редакцию 13 июня 2020

Аннотация. Цель работы – разработка моделей и алгоритмов оптимизации паросочетаний в динамически формируемых графах асимметричных отношений в координируемых открытых системах взаимодействующих агентов с централизованным и коллективным управлением. Динамическая асимметричная задача оптимизации паросочетаний здесь возникает как результат компромиссной аппроксимации отображения метода динамического программирования на поток известных открытых задач о назначении или нескольких странствующих коммивояжёров. Однако представленные таким образом альтернативы ветвления на независимых задачах не учитывают взаимозависимость реальных отношений между агентами и их заданиями, включая их привязку ко времени. Игнорирование зависимости альтернатив ветвления приводит к задержке момента или потере качества назначения заданий координируемым агентам. Основная идея предлагаемой реализации известного для эффективного управления принципа – откладывание момента принятия окончательного решения на наиболее поздний момент, учет восприимчивости системы к локальным изменениям переменных состояния. Взаимозависимость состояний выявляется на основе анализа соответствия графа текущего паросочетания оптимальному решению на подграфе совершенного паросочетания. Переход между состояниями реализуется инкрементальной версией алгоритма реоптимизации решения линейных задач о назначении методом кратчайшего пополняющего пути. Пространство состояний поиска – динамически формируемый двудольный разреженный граф альтернатив сочетания агентов и задач, представленный списком дуг. Для выделения множеств изменившихся дуг предложено дополнить веса дуг границами интервалов устойчивости решения, факультативно формируемых в фоновом режиме. По умолчанию вес измененной дуги совпадает с границей интервала устойчивости. На каждом цикле коррекции списков агентов, задач и их ассоциаций выделяются подмножества элементов, для которых требуется пересмотр паросочетания. Усиленное условие отбора таких элементов – выход за границы интервала устойчивости. При этом асимметрия задачи назначения учитывается выбором структуры смежности для доли графа с минимумом вершин. В результате время реакции процедур решения задачи назначения сокращается на порядок.

Ключевые слова: метод кратчайшего пополняющего пути, динамическая задача о назначении.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Ревотюк М.П., Хаджинова Н.В., Кузнецов А.П., Шилин Л.Ю. Динамическая асимметричная задача о назначении в открытых многоагентных системах. Доклады БГУИР. 2020; 18(5): 53-61.

DYNAMIC ASYMMETRIC ASSIGNMENT PROBLEM IN OPEN MULTI-AGENT SYSTEMS

MIKHAIL P. REVOTJUK, NATALIA V. KHAJYNOVA, ALEXANDER P. KUZNETSOV, LEONID Y. SHILIN

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Submitted 13 June 2020

Abstract. The purpose of the work is to develop models and algorithms for optimizing matching in dynamically generated graphs of asymmetric relations in coordinated open systems of interacting agents with centralized and collective control. The dynamic asymmetric matching optimization problem arises here as a result of a compromise approximation of the mapping of the dynamic programming method onto a stream of known open assignment problems or several traveling salesmen. However, the branching alternatives presented in this way for independent tasks do not take into account the interdependence of real relationships between agents and their tasks, including their relationship to time. Ignoring the dependence of branching alternatives leads to a delay in the moment or to a loss in the quality of assignment of tasks to coordinated agents. The main idea of the proposed implementation of the principle known for effective control is to postpone the moment the final decision is made to the latest moment, taking into account the susceptibility of the system to local changes in state variables. The interdependence of states is revealed on the basis of the analysis of the correspondence of the graph of the current matching with the optimal solution on the subgraph of perfect matching. The transition between states is implemented by the incremental version of the reoptimization algorithm for solving linear problems of assigning the shortest replenishing path using the method. The space of search states is a dynamically generated bipartite sparse graph of alternatives for a combination of agents and tasks, represented by a list of arcs. To highlight the sets of changed arcs, it is proposed to supplement the weight of the arcs with the boundaries of the stability intervals of the solution, optionally formed in the background. By default, the weight of the modified arc matches the boundary of the stability interval. On each correction cycle of the lists of agents, tasks, and their associations, subsets of elements are selected for which reconsideration of matching is required. An enhanced condition for the selection of such elements is to go beyond the boundaries of the stability interval. In this case, the asymmetry of the assignment problem is taken into account by choosing the adjacency structure for the fraction of the graph with a minimum of vertices. As a result, the reaction time of procedures for solving the assignment problem is reduced by an order of magnitude.

Keywords: shortest augmenting path method, dynamic assignment problems.

Conflict of interests. The authors declare no conflict of interests.

For citation. Revotjuk M.P., Khajynova N.V., Kuznetsov A.P., Shilin L.Y. Dynamic asymmetric assignment problem in open multi-agent systems. Doklady BGUIR. 2020; 18(5): 53-61.

Введение

Объект исследования – процессы управления системами взаимодействующих агентов. Под агентом будем понимать сущность, предназначенную для решения некоторых задач. Подобные системы в последнее время получают широкое распространение в промышленности, транспорте и других отраслях, характеризуются автономностью поведения агентов [1]. Основная функция управления в таких системах – координация взаимодействия агентов для обеспечения требуемых отношений в потоках решаемых задач.

Централизованный и коллективный подход к управлению – потенциально наиболее эффективные, но требуют решения задач повышенной вычислительной сложности [2]. Компромисс между точностью и своевременностью решения заключается в использовании различного рода аппроксимаций. Наиболее практичным оказывается прием отображения процедур оптимизации многошаговых дискретных процессов в системах агентов на множество линейных задач о назначении. При этом веса связи агентов и задач представляют скалярную свертку локальных критериев с учетом неопределенности будущих событий. Некоторые параметры весов зависят от состояния системы, а в процессе построения дерева поиска оцениваются методом имитационного моделирования [2].

В ряде процедур оптимизации управления, например, методами динамического программирования, ветвей и границ, порождаемые варианты решения отличаются незначительными изменениями локальных переменных состояния. Очевидно, что отражение последствий таких изменений на целевую функцию приведет к ускорению поиска оптимального решения. Последнее является побудительным мотивом изучения способов формализации и решения задачи построения оптимального паросочетания агентов и решаемых ими задач с учетом возможности откладывания принятия окончательного решения на наиболее поздний момент.

Обсуждаемая далее динамическая асимметричная задача о назначении рассматривается в контексте потока событий координатора открытых многоагентных систем, решающих предопределенные общей целью взаимосвязанные во времени задачи. Динамика здесь соответствует появлению моментов формирования назначения агентов, необходимых и достаточных для решения целевого потока задач к определенному сроку. Неопределенность на этапе формирования назначения информации о доступных агентах и учет предложений о готовности к решению задач в будущем создает асимметрию отношений «агент – задача» на этапах выделения альтернатив поиска решений.

Методика решения задачи

Модель динамической задачи назначения. Построение оптимального паросочетания агентов из множества A и решаемых ими задач из множества T может быть выполнено в результате решения хорошо изученной линейной задачи о назначении (ЛЗН): для заданных оценок эффективности назначения $C: T \times A \to R^{+}$ найти паросочетание $X: T \to A$ минимального веса [3]. Пусть $|A| = m,$ а $|T| = n.$ В случае, когда $m = n,$ ЛЗН в матричной формулировке

$$\min\left\{ \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} \,\middle|\, \sum_{i=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1;\ x_{ij} \ge 0;\ i, j = \overline{1,n} \right\} \tag{1}$$

может считаться рутинной задачей этапа реализации вычислительной схемы алгоритма, когда так называемые открытые ЛЗН [3, 4]

$$\min\left\{ \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} \cdot x_{ij} \,\middle|\, \sum_{i=1}^{m} x_{ij} = \sum_{j=1}^{n} x_{ij} = 1;\ x_{ij} \ge 0;\ i = \overline{1,m},\ j = \overline{1,n} \right\} \tag{2}$$

становятся частным случаем задачи (1) после формального дополнения матрицы C до размера $\max(m, n)$ нулевыми элементами. В случае необходимости поиска паросочетания $X: T \to A$ максимального веса, в выражении (1) достаточно произвести замену $c_{ij} \leftarrow c^{*} - c_{ij},\ i, j = \overline{1,n},$ где $c^{*} = \max\{c_{ij},\ i, j = \overline{1,n}\}.$

Отрытые ЛЗН, решаемые процедурами без явного преобразования в (1) [4, 5], всегда следует рассматривать в условиях, когда $m \le n$. Случай $m > n$ для отказа от необходимости перебора сочетаний строк легко сводится к решению (2) после реального или виртуального транспонирования матрицы C.

Вычислительная сложность решения ЛЗН с помощью одного из лучших среди известных модифицированного метода кратчайшего пополняющего пути (Shortest Augmenting Path, SAP) при применении алгоритма Дейкстры с кучей Фибоначчи имеет асимптотики от $O(n^{3/4} m \cdot \log c^{*})$ до $O(\sqrt{n} m \cdot \log(nc^{*}))$ [3]. Отсюда следует, что время решения открытой ЛЗН зависит прямо пропорционально от количества строк. Такая зависимость сохраняется и в случае реоптимизации назначения после коррекции некоторых строк матрицы статической ЛЗН [5].

Очевидно, что последнее утверждение открывает путь сокращения вычислительной сложности решения последовательно порождаемых ЛЗН, если учесть реально существующие связи между их матрицами. Отображение $X:T \to A$, формируемое в результате оптимизации паросочетания, предполагает рассмотрениие связи задач и агента на состояниях «назначена – не назначена». В реальном времени множество перечисленных состояний предлагается дополнить состоянием «не рассмотрена», отражая появление в произвольный момент новых альтернатив связи задач и агента по отношению к оптимальному парасочетанию. Это позволяет отразить возможность согласованного представления формируемых отношений задач T и агентов A с представлением оптимального парасочетания. Согласование таких представлений будем проводить на основе выбора для определения исходных данных ЛЗН и пространства поиска решения динамически формируемых графов паросочетаний в форме списка дуг. Новые альтернативы связи задач и агента в момент их появления обуславливают необходимость пересмотра парасочетаний, решая ЛЗН с обновленными данными.

Очевидно, что в случае решения потока зависимых ЛЗН полный пересчет очередного варианта ЛЗН, например, венгерским методом, потребует $O(n^3)$ операций. Однако итерация учета изменений любой отдельной строки имеет вычислительную сложность $O(n^2)$ [5, 6]. После изменения Δm строк сложность пересчета – $O(\Delta m \cdot n^2)$, что характеризует полезность наследования результатов предшествующего решения. Наследование результатов улучшает реактивность системы координации, даже если до момента принятия решения о назначении не имеется полной информации об окончательной подлежащей решению задаче. Далее, на основе анализа оценок жесткости элементов парасочетаний будут рассмотрены приемы выделения минимально необходимого для точного решения очередной ЛЗН множества измененных строк.

Идея наследования результатов предшествующего решения неявно предлагалась в версии алгоритма решения ЛЗН на основе венгерского метода для случая, когда меняются элементы матрицы стоимости [6]. Известно, что наиболее эффективные для решения задачи (2) алгоритмы на основе метода пополняющего пути и их модификации учитывают особенности двойственной задачи

$$\max\left\{\sum_{i=1}^{m} u_i + \sum_{j=1}^{n} v_j \;\middle|\; c_{ij} - u_i - v_j \ge 0,\ i=\overline{1,m},\ j=\overline{1,n}\right\}. \tag{3}$$

В этом случае наследуемые результаты решения открытых ЛЗН включают:

сопряженные векторы назначений столбцов матрицы ее строкам и строк матрицы ее столбцам – $R=\{r_j=i \,|\, x_{ij}=1,\ j=\overline{1,n}\}$ и $Q=\{q_i=j \,|\, x_{ij}=1,\ i=\overline{1,m}\}$;

векторы потенциалов строк и столбцов – $U=\{u_i,\ i=\overline{1,m}\}$ и $V=\{v_j,\ j=\overline{1,n}\}$.

Наличие пар векторов формально создает избыточность наследуемых результатов:

$r_{q_i}=i,\ i=\overline{1,m}$ и $q_{r_j}=j,\ j=\overline{1,n}$ – по определению, но их наличие позволяет исключить операции поиска альтернатив расширения пополняющих путей;

$u_i+v_j=c_{ij}, i=\overline{1,m},\ j=\overline{1,n}$ – известное условие жесткости элементов оптимального паросочетания (в методе SAP используется для сокращения размера пространства состояния).

Указанная избыточность позволяет обеспечить выбор направления отображения $A \leftrightarrow T$, включая виртуальное транспонирование матрицы C, после выявления нарушения условия $m \le n$. Переключение направления отображения посредством замены ссылок исключает операции копирования переменных состояния решения последовательно возникающих ЛЗН. Сложность подобной операции $-O(m+n)$, что в ряде случаев может стать практически значимым при большом количестве атрибутов отношения $A \leftrightarrow T$ [1, 2].

Алгоритм динамического назначения. Предлагается для оптимизации динамического назначения модифицировать известный инкрементальный алгоритм решения статических ЛЗН [6]. Модификация состоит в замене оригинальной процедуры итерации венгерского метода процедурой расширения кратчайшего пополняющего пути метода SAP и применении такой процедуры после добавления новой или коррекции существующей строки $i,\ i \in \overline{1,m}$ текущего паросочетания [5, 7]. Для случая решения динамической ЛЗН вида (3) эвристический этап инициализации начального назначения, обычно предлагаемый многими авторами для решения (1) [3], может быть исключен (рис. 1).

```
function  SLAP() {
  foreach j ∈ 1,n  { r_j = 0,  v_j = 0; }
  foreach i ∈ 1,m  sap(i);
}
```

Рис. 1. Алгоритм решения отрытой ЛЗН методом SAP
Fig. 1. The algorithm for solving Open Linear Assignment Problem using the SAP method

Обозначим для каждой вершины x графа множество смежных выходных вершин x', $x' = \{y \mid x \to y\}$, а входных вершин – $'x$, $'x = \{x \mid x \to y\}$, где $x \to y,\ x, y \in A \cup T$. Итерации назначения строк методом SAP реализует функция *sap* (рис. 2). Такая функция обеспечивает формирование наследуемых на других этапах векторов $U,\ V,\ Q$ и R.

```
function sap(i)  {
  foreach j ∈ i'  { p_j = i, d_j = c_ij − v_j; }
  K = ∅,  S = ∅, T = {i'};
  while (true)  {
   if  (|S| = 0) {
    h = min{ d_j | j ∈ T};
    S = { j | (d_j = h) ∧ ( j ∈ T)}; T ← T \ S;
    foreach  j ∈ S  if  (r_j = 0)  go to back;
   }
   k = S_1,  S ← S \ {k},  K ← K ∪ {k},  l = r_k ;
   foreach j ∈ T   if (h + c_lj − v_j < d_lj)  {
     d_j = h + c_lj − v_j;  p_j = l;
     if  (d_j = h) {
      if  (r_j = 0)  go to backtrack;
      S ← S ∪ { j},  T ← T \ { j};
     }
   }
  backtrack:  foreach  k ∈ K  v_k ← v_k + d_k − h;
  do {l = p_j; r_j = l; k = j; j = q_l; q_l = k; } while (i ≠ l);
 }
}
```

Рис. 2. Алгоритм итерации назначения методом SAP
Fig. 2. Algorithm of SAP Assignment Iteration

Графы реальных ЛЗН чаще всего разреженные, а с учетом удобства отражения их изменчивости во времени предпочтительная форма их представления – список дуг. Изменение таких графов также будем представлять списком дуг. В результате объединения на очередном шаге k получаем новый граф, в котором множество исходных вершин представляет множество агентов A^k, а множество конечных вершин – множество подлежащих решению задач T^k.

Пусть на любом этапе k имеем $m = |A^k|$, а $n = |T^k|$. Очевидно, что истинность условия $m \le n$ определяет необходимость поиска паросочетания на задаваемом структурой смежности отображении $FS: A \to T,\ FS = \{(x, x'),\ x \in A\}$, а в противном случае – на отображении $BS: T \to A,\ BS = \{(y, 'y),\ y \in T\}$. Отображения FS и BS создаются для формирования соответствующих отношений на списке дуг $(x, y, c_{xy}),\ x \in A,\ y \in T$. В любом случае ключевые элементы выбранного отношения будут соответствовать строкам открытой ЛЗН (2). Итерации назначения (рис. 2) в случае $m \le n$ необходимо выполнять по элементам $\{U, R, FS\}$. В случае $m > n$ роль строк должны исполнять элементы $\{V, Q, BS\}$.

Обозначим G^k – заданный в форме списка дуг граф допустимых на этапе k сочетаний агентов и задач, где дуги представляют тройки (x, y, c_{xy}), $x \in A^k$, $y \in T^k$.

Пусть ΔG^k – список изменяемых, добавляемых или удаляемых дуг, структура которого совпадает со структурой графа G^k. На следующем шаге новый список получим как результат операции $G^{k+0} = G^k \cup \Delta G^k$, где объединение выполняется с группированием пар (x, y) с целью однозначности ассоциации $x \leftrightarrow y$. Отсюда следует алгоритм выделения подлежащих реоптимизации строк, формирующий стек K номеров таких строк на этапе выполнения операции присваивания новых весов $\Delta m = |K|$ дуг:

$$\begin{cases} K = \varnothing; \\ forearch((i, j, c_{ij}) \subset \Delta G^k)\ \{ \\ \quad if\ (c_{ij}^k \ne c_{ij})\ \{ \\ \quad\quad if\ (I(i) \ne k)\ \{ \\ \quad\quad\quad K \leftarrow K \cup \{i\},\ I(i) \leftarrow k; \\ \quad\quad \} \\ \quad\quad c_{ij}^{k+1} \leftarrow c_{ij}; \\ \quad \} \\ \} \\ forearch(i \in K)\ sap(i); \end{cases} \tag{4}$$

Изменяемые строки помечены номером этапа k в глобальном массиве I, поэтому вычислительная сложность выделения строк – $O(\Delta m)$. Алгоритм (4) можно дополнить операциями выделения измененных столбцов в соответствующем глобальном массиве J.

Фильтрация несущественных событий. Можно заметить, что анализ в (4) лишь фактов изменения весов дуг можно дополнить количественной оценкой влияния такого изменения на существующее на этапе k паросочетание, если для каждой дуги получить интервалы устойчивости оптимального решения [7, 8]. Очевидно, что контроль на этапе $k+0$ условия выхода веса дуг за границы – $c_{ij}^{k+0} \notin \{(a_{ij}^k, b_{ij}^k), i = \overline{1,m},\ j = \overline{1,n}\}$ – позволяет легко распознать необходимость перестройки паросочетаний агентов из множества $A_k = \{i \mid c_{ij}^k < \infty,\ j = \overline{1,n}\}$ и задач из множества $T_k = \{j \mid c_{ij}^k < \infty, i = \overline{1,m}\}$. Элементы $A_k \times T_k$ определяют область возмущения текущего решения, которая должна быть учтена процедурой координации системы агентов [2].

Покажем далее, что наличие таких интервалов сокращает размерность стека K, не меняет порядок вычислительной сложности реоптимизации, а их определение может выполняться в фоновом режиме. По умолчанию интервал устойчивости тривиально отображает существующий вес дуг – $(a_{ij}^k = c_{ij}^k,\ b_{ij}^k = c_{ij}^k),\ i = \overline{1,m},\ j = \overline{1,n}$.

Пусть на этапе k выделены ребра графа оптимального паросочетания $E_m = \{(r_j, j) \mid r_j < n,\ j = \overline{1,n}\}$, а оставшиеся ребра возможных паросочетаний обозначены как $E_u = \{(i, j),\ i = \overline{1,m},\ j = \overline{1,n}\} \setminus E_m$. По определению $E_m \cap E_u = \varnothing$ и $E_m \cup E_u = E$. Отсюда следует, что $|\{(a_{ij}^k, b_{ij}^k),\ i = \overline{1,m},\ j = \overline{1,n}\}| = |E|$.

Обозначим $\delta_m(x, y)$ и $\delta_u(x, y)$ – допустимое изменение веса любого ребра $x \to y$ во множествах соответственно E_m и E_u. Интервалы допустимых изменений значений весов ребер, в которых результат оптимизации назначения остается неизменным, для ЛЗН вида (1) просто вывести из элементарных рассуждений:

$$I_{\min}^m(x, y) = (-\infty, c_{xy} + \delta_m(x, y)],\ (x, y) \in E_m,\quad I_{\min}^u(x, y) = (c_{xy} - \delta_u(x, y), +\infty],\ (x, y) \in E_u.$$

Действительно, ребро графа $x \to y$ покинет оптимальное паросочетание, если его вес увеличить назначением любого значения из интервала $(c_{xy} + \delta_m(x, y), +\infty)$.

Идея определения $\delta_m(x, y)$ основана на известном для элементов оптимальном условии жесткости $c_{xy} = u_x + v_y$, $(x, y) \in E_m$. Пусть элементы выражения для оценки текущего оптимального решения ЛЗН выделены верхним индексом m:

$$Z^m = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}^m x_{ij}^m = \sum_{i=1}^{m} u_i^m + \sum_{j=1}^{n} v_j^m. \qquad (5)$$

Нетрудно показать, что если изменить вес ребра $x \to y, (x, y) \in E_m$, то итерации процесса реоптимизации на основе метода пополняющего пути (рис. 2), начинающиеся в вершине x, всегда завершатся в вершине y [5]. При этом меняется лишь потенциал строки x, а в соответствии с (5) $u_x^u - u_x^m = Z^u - Z^m$. Здесь верхним индексом u отмечен факт исключения ребра $x \to y$ из оптимального паросочетания после замены $c_{xy}^m \leftarrow c_{xy}^u = \infty$. Отсюда следует, что $\delta_m(x, y) = Z^u - Z^m$, а интервал безопасного изменения значений веса назначенного ребра:

$$I_{\min}^m(x, y) = (-\infty, c_{xy} + Z^u - Z^m] = (-\infty, c_{xy} + u_x^u - u_x^m], \ (x, y) \in E_m. \qquad (6)$$

Суммирование приращений потенциалов строк для всех этапов k исключает необходимость прямолинейного вычисления оценок решений задачи (1), требующего $m+n$ шагов. Последнее важно для эффективной реализации операций отсечения альтернатив поиска в процедурах, например, прямого динамического программирования.

Рассмотрим далее ребра, не принадлежащие оптимальному решению, когда $c_{xy} \le u_x + v_y$, $(x, y) \in E_u$. Если вес таких ребер менять в интервале $(u_x + v_y, +\infty)$, то структура решения (2) остается неизменной. Используя линейность целевой функции (2), для определения $\delta_u(x, y)$ можно воспользоваться выражением (6). Предлагается просто поменять направление шагов процесса, рассматривая условие включения ребра $x \to y$ в оптимальное паросочетание. Начальный шаг в (6) становится решением ЛЗН с обязательным для назначения ребра $x \to y$ значением веса $c_{xy}^u \leftarrow c_{xy}^m = -\infty$, а конечный шаг будет соответствовать решению ЛЗН с исходной матрицей. Таким образом, получаем

$$I_{\min}^u(x, y) = (-\infty + Z^m - Z^u, +\infty] = (-\infty + u_x^m - u_y^u, +\infty], \ (x, y) \in E_u. \qquad (7)$$

Результат итераций метода SAP (рис. 2) содержит лишь значение потенциала столбца, но пригоден для определения $I_{\min}^m(x, y)$ и $I_{\min}^u(x, y)$, так как значение потенциала строки можно представить в виде $u_x = v_y - c_{xy}$, $(x, y) \in E_m$. Очевидно, что так как $|E_m| = m$, а $|E_u| = m \cdot (n-1)$, то вычислительная сложность реализации (6) и (7) – $O(n^4)$.

Определение интервалов устойчивости может проводиться в фоновом режиме. При этом, назначая порядок рассмотрения отдельных дуг, целесообразно использовать в качестве оценки относительного приоритета значения их веса. На переходных этапах оптимизации паросочетаний необходимо использовать (5). При наличии интервалов устойчивости условие $(c_{ij}^k \ne c_{ij})$ в (4) может быть усилено проверкой условия $c_{ij} \notin (a_{ij}, b_{ij})$, что снижает сложность перестройки решения.

Отказ от перестройки паросочетаний возможен при появлении новой альтернативы связи $x \to y$, когда на этапе k выполняется $(x \notin A^k) \vee (y \notin T^k)$. Такая связь не обязательно должна рассматриваться, если $c_{xy} > \min(b_{iy}, i \in' y)$ или $c_{xy} > \min(a_{xj}, j \in x')$.

Результаты и их обсуждение

Рассмотренная задача о динамическом назначении отражает дискретность процесса формирования портфелей предложенных услуг и заявок на обслуживание в многоагентных системах, используя жадный инкрементальный алгоритм решения задачи упреждающего поиска окончательного назначения. Очевидно, что в худшем случае, когда приходится изменять граф оптимального паросочетания при поступлении последней группы новых заявок, задержка времени принятия окончательного решения определяется лишь сложностью ее обработки. Как следствие, система координации решения задач способна до последнего момента учесть вновь возникающие альтернативы назначения. Такая возможность становится привлекательной в системах безропотных агентов с беспилотным управлением.

Предложенные структуры данных представления пространства состояния, алгоритмы локализации области возмущений и оценки интервалов устойчивости позволяют снизить вычислительную сложность обработки последствий перехода в новое состояние до линейной зависимости от количества изменившихся ассоциаций задач и агентов. Последнее становится важным при использовании ЛЗН для релаксации на итерациях процедур, например, динамического программирования, ветвей и границ, простого или направленного перебора [2–4, 6]. Время проведения отдельных итераций экспоненциально сложных комбинаторных алгоритмов иногда можно сократить отображением зависимости решаемых агентами задач на потоки порождаемых ЛЗН. Однако при этом следует контролировать нарушения принципа независимости ЛЗН от предыстории.

Заключение

Полученные модели и алгоритмы поиска и пересмотра оптимального паросочетания определяют границы области возмущения текущего назначения агентов подлежащим решению задачам. Их наличие позволяет усилить логические условия отказа от итераций пересмотра решения при несущественных изменениях параметров внешней среды, что ускоряет реакцию системы управления на обработку информации о состоянии агентов и задач, обеспечивая полноту учета альтернатив оптимизации назначения в реальном времени.

Список литературы

1. Korsah G.A., Stentz A., Dias M.B. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*. 2013;12(32):1495-1512.
2. Spivey M.Z., Powel W.B. The Dynamic Assignment Problem. *Transportation science.* 2004;38(4):399-419.
3. Pentico D.W. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*. 2007;176(2):774-793.
4. Bijsterbosch J., Volgenant A. Solving the Rectangular assignment problem and applications. *Annals of Operations Research*. 2010;181(1):443-462.
5. Ревотюк М.П., Батура П.М., Полоневич А.М. Реоптимизация решения задач о назначении. *Доклады БГУИР*. 2011;1(55):55-62.
6. Toroslu I.H., Üçoluk G. Incremental assignment problem. *Information Science*. 2007;177:1523-1529.
7. Ревотюк М.П., Кароли М.К., Батура П.М. Реализация метода ветвей и границ для решения задач коммивояжера с разреженными матрицами. *Доклады БГУИР*. 2013;7(77):25-31.
8. Ревотюк М.П., Кароли М.К., Батура П.М. Быстрая оценка интервалов устойчивости решения линейных задач о назначении. *Доклады БГУИР*. 2013;5(75):30-36.

References

1. Korsah G.A., Stentz A., Dias M.B. A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*. 2013;12(32):1495-1512.
2. Spivey M.Z., Powel W.B. The Dynamic Assignment Problem. *Transportation science.*2004;38(4):399-419.
3. Pentico D.W. Assignment problems: A golden anniversary survey. *European Journal of Operational Research*. 2007;176(2):774-793.

4. Bijsterbosch J., Volgenant A. Solving the Rectangular assignment problem and applications. *Annals of Operations Research*. 2010;181(1):443-462.
5. Revotjuk M.P., Batura P.M., Polonevich A.M. [Reoptimization of the linear assignment problem]. *Doklady BGUIR=Doklady BSUIR*. 2011;1(55):55-62. (In Russ.)
6. Toroslu I.H., Üçoluk G. Incremental assignment problem. *Information Science*. 2007;177:1523-1529.
7. Revotjuk M.P., Qaraleh M.K., Batura P.M. [Implementation the branch and bound method for solving the traveling salesman problem with sparse matrix]. *Doklady BGUIR = Doklady BSUIR.* 2013;7(77):25-31. (In Russ.)
8. Revotjuk M.P., Qaraleh M.K., Batura P.M. [Quick evaluation of the interval stability of the linear assignment problem solutions]. *Doklady BGUIR = Doklady BSUIR.* 2013;5(75):30-36. (In Russ.)

Вклад авторов

Все авторы в равной степени внесли вклад в написание статьи.

Authors contribution

All authors equally contributed to the writing of the article.

Сведения об авторах

Ревотюк М.П., к.т.н., доцент, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники.

Хаджинова Н.В., старший преподаватель кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники.

Кузнецов А.П., д.т.н., профессор, профессор кафедры систем управления Белорусского государственного университета информатики и радиоэлектроники.

Шилин Л.Ю., д.т.н., профессор, декан факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники.

Information about the authors

Revotjuk M.P., PhD, Associate Professor, Associate Professor of Information Technologies in Automated Systems Department of Belarusian State University of Informatics and Radioelectronics.

Khajynova N.V., Senior Lecturer of Information Technologies in Automated Systems Department of Belarusian State University of Informatics and Radioelectronics.

Kuznetsov A.P., D.Sci., Professor, Professor of Control Systems Department of Belarusian State University of Informatics and Radioelectronics.

Shilin L.Y., D.Sci., Professor, Dean of Faculty of Information Technologies and Control of Belarusian State University of Informatics and Radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, 6,
Белорусский государственный университет
информатики и радиоэлектроники
тел. +375-17-293-86-58;
e-mail: rmp@bsuir.by
Ревотюк Михаил Павлович

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka str., 6,
Belarusian State University
of Informatics and Radioelectronics
tel. +375-17-293-86-58;
e-mail: rmp@bsuir.by
Revotjuk Mikhail Pavlovich