Моделирование и анализ данных
2019. Том 09. № 4. С. 100–111
doi: 10.17759/mda.2019090408
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2019. Vol. 09, no. 4, pp. 100–111
doi: 10.17759/mda.2019090408
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

УДК 519.85

Мультиагентное моделирование в задачах формирования расписаний

Сивакова Т.В.*
ИПМ им. М.В.Келдыша РАН, Москва, Россия
РЭУ им. Г.В. Плеханова, Москва, Россия
sivakova15@mail.ru

Судаков В.А.**
Московский авиационный институт, Москва, Россия
ИПМ им. М.В.Келдыша РАН, Москва, Россия
sudakov@ws-dss.com

В статье исследуется применение мультиагентных технологий для решения оптимизационных задач. Показано, как мультиагентные системы позволяют работать с ограничениями в распределенной вычислительной среде. Формализована задача составления расписания. Разработано программное обеспечение и проведены вычислительные эксперименты, показавшие эффективность предложенного подхода.

Ключевые слова: мультиагентные системы, предпочтения агентов, оптимизация, распределенные системы.

Введение

Рассмотрим проблему распределения множества задач на множество ресурсов (например, планирование лекций для аудиторий или пакета заданий для нескольких процессоров). Это распространенная и важная проблема, которую можно формализовать, используя мультиагентный подход.

Для цитаты:
Сивакова Т.В., Судаков В.А. Мультиагентное моделирование в задачах формирования расписаний // Моделирование и анализ данных. 2019. Том 09. № 4. С. 100–111. doi: 10.17759/mda.2019090408

***Сивакова Татьяна Владимировна,** научный сотрудник, Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук»(ИПМ им. М.В.Келдыша РАН), научный сотрудник, Российский экономический университет имени Г.В. Плеханова (РЭУ им. Г.В. Плеханова), Москва, Россия. E-mail: sivakova15@mail.ru

****Судаков Владимир Анатольевич,** доктор технических наук, профессор, Московский авиационный институт (национальный исследовательский университет), ведущий научный сотрудник, Федеральное государственное учреждение «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук»(ИПМ им. М.В.Келдыша РАН), Москва, Россия. E-mail: sudakov@ws-dss.com

Обработку ограничений можно рассматривать как широкую и разнообразную область исследований, объединяющую методы и алгоритмы, которые охватывают многие различные дисциплины, включая исследование операций, компьютерное зрение, искусственный интеллект и теорию принятия решений. Все эти области имеют дело со сложными проблемами, которые можно сделать более понятными, тщательно продумав ограничения, определяющие структуру проблемы.

В данной статье показано, как обработка ограничений может использоваться для решения проблем оптимизации в мультиагентных системах (MAS – multi-agent system). Рассматриваются распределенные подходы к задачам оптимизации с ограничения (DCOP – distributed constraint optimization). В DCOP набор агентов должен прийти к какому-то соглашению (обычно через какую-то форму переговоров), о том, какие действия должен предпринять каждый агент, чтобы совместно получить наилучшее решение для всей системы [1]. Эта структура успешно используется не только для планирования собраний, но и в сенсорных сетях, где датчики должны договориться о том, на какую цель они должны ориентироваться, чтобы получить наиболее точную оценку местоположения целей. Для тестирования эффективности алгоритмов решения DCOP, часто используют задачи проверки выполнимости булевых функций и раскраски графов. Общим ключевым аспектом DCOP для MAS является то, что каждый агент ведет локальные переговоры только с подмножеством других агентов (обычно называемых соседями), которые могут непосредственно влиять на его поведение. В зависимости от постановки задачи и используемой методики решения этот аспект может значительно сократить вычислительные усилия, с которыми сталкивается каждый агент, что делает сложные проблемы доступными даже для крупномасштабных систем. Так в задаче планирования собраний, агент будет напрямую вести переговоры только с теми агентами, с которыми он должен встретиться, что обычно составляет небольшое подмножество агентов, вовлеченных во всю проблему.

Наряду с мультиагентными подходами, сейчас активно развиваются метаэвристические методы оптимизации, которые хотя и не гарантируют нахождения точного решения, но обычно позволяют найти приемлемое решение за приемлемое время [2].

Возможная формализация DCOP для задачи планирования собрания включает набор агентов, представляющих людей, участвующих в собрании, и набор переменных, которые представляют возможное время начала данного собрания в соответствии с участником. Ограничения предписывают равенство переменным, представляющим время начала одного и того же собрания разных агентов, и гарантируют, что переменные, которые представляют время начала разных встреч одного и того же агента не совпадает. Наконец, предпочтения могут быть представлены как мягкие ограничения на время начала встреч, и общая цель состоит в том, чтобы оптимизировать сумму всех мягких ограничений. Хотя в этом параметре у нас есть личные предпочтения, мы максимизируем сумму предпочтений всех агентов, и, таким образом, рассматривается сценарий, когда агенты полностью сотрудничают, то есть они готовы уменьшить свою собственную локальную полезность, если это максимизирует глобальную полезность [3].

Постановка задачи

В качестве конкретного примера применения DCOP рассмотрим задачу формирования расписания авиарейсов.

Введем обозначения:

$i = 1..n$ – номера типов летальных аппаратов (ЛА),

v_i – вместимость ЛА,

c_i – стоимость ЛА i-го типа,

q_i – стоимость часа полета ЛА i-го типа,

w_i – постоянные затраты на рейс ЛА i-го типа (независящие от продолжительности рейса),

s_i – максимально возможное число ЛА i-го типа,

j – номер аэропорта,

$A = \left\| a_{j_1 j_2} \right\|$ – объем перевозки (пассажиров в неделю),

$a_{jj} = 0$,

j_1 – аэропорт вылета,

j_2 – аэропорт прилета,

$$s_i = \sum_{j_1=1}^{m} \sum_{j_2=j_1+1}^{m} \left\lceil \frac{\max\left(a_{j_1 j_2}, a_{j_2 j_1}\right)}{7 v_i} \right\rceil,$$

K – максимально-возможное число ЛА, которые могут потребоваться для перевозки:

$$K = \sum_{i=1}^{n} s_i;$$

$k = 1..K$ – номера конкретных ЛА (они могут не все в дальнейшем потребоваться),

$i = f(k)$ – перевод индекса конкретного ЛА в индекс типа ЛА,

$D_k = \left\| d_{j_1 j_2}^k \right\|$ – средняя продолжительность рейса (в часах), зависит от типа ЛА (в общем случае матрица не симметричная),

$x_{lw}^k \in \{0,1\}$ – оптимизационная переменная, равна единице, если самолет k совершает рейс l в день w,

$l = 1..L$ – номер потенциального рейса (все номера рейсов могут быть не задействованы). Это уникальный номер рейса в один день.

Номер рейса l связан взаимно-однозначным соответствием с кортежем $(j_1, j_2, u_{j_1 j_2})$. При этом следует брать только те пары j_1, j_1 для которых требуются перевозки: $a_{j_1 j_2} \neq 0$.

Обозначим: отображение $l \to j_1$ как функцию $j_1 = g_1(l)$, а отображение $l \to j_2$ как функцию $j_2 = g_2(l)$,

w – номер дня (если составляем недельное расписание $w = 1..7$, но вместо 7 может быть и другое число),

$u_{j_1 j_2} = 1..U_{j_1 j_2}$ – это порядковый номер рейса для любого дня,

$U_{j_1 j_2}$ – максимальное кол-во рейсов, которые нам могут потребоваться за один день в данном направлении в наихудшем случае (недельный трафик за один день ЛА наименьшей вместимости):

$$U_{j_1 j_2} = \left\lfloor \frac{a_{j_1 j_2}}{\min_i v_i} \right\rfloor,$$

L – максимальное число рейсов в день, которые возможно придётся задействовать.

$$L = \sum_{j_1=1}^{m} \sum_{j_2=1}^{m} U_{j_1 j_2},$$

t_l – время вылета рейса. Оптимизационная переменная. Время считается с заданным шагом дискретизации на всем периоде планирования;

$\tau_l^k = t_l + d_{j_1 j_2}^k$ – время прилета,

d_j – ограничение на пропускную способность ВПП аэропорта j. Используется в соответствующих ограничениях:

$$\forall k_1 \neq k_2 \ \forall l_1 \neq l_2 \ \forall w_1, w_2 \forall j \ (\left(x_{l_1 w_1}^{k_1} = x_{l_2 w_2}^{k_2} = 1\right) \to \left(\left(g_1(l_1) = g_1(l_2) = \text{j}\right) \to \left|t_{l_1} - t_{l_2}\right| \geq d_j\right) \&$$

$$\left(\left(g_2(l_1) = g_1(l_2) = \text{j}\right) \to \left|\tau_{l_1}^k - t_{l_2}\right| \geq d_j\right) \& \left(\left(g_2(l_1) = g_2(l_2) = \text{j}\right) \to \left|\tau_{l_1}^{k_1} - \tau_{l_2}^{k_2}\right| \geq d_j\right));$$

p_j – пропускная способность аэропорта j пассажиров в сутки, позволяет записать ограничение:

$$\forall w \forall j \left(\sum_{l \in \{l: g_1(l) = j \vee g_2(l) = j\}} \sum_k v_{f(k)} x_{lw}^k \leq p_j \right).$$

Оптимизационная переменная:

y_k – базовый аэропорт для k-го самолета,

$$\forall k \forall l \forall w \left(\left(x_{lw}^k = 1\right) \to \left(\left(g_2(l) = y_k\right) \vee P(k,l)\right) \& \left(\left(g_1(l) = y_k\right) \vee Q(k,l)\right)\right),$$

$$P(k,l) = \exists l_* \exists w \begin{bmatrix} \left(x_{l_* w}^k = 1\right) \& \left(g_2(l) = g_1(l_*)\right) \& \left(\mu_k \leq t_{l_*} - \tau_l^k \leq \eta_{g_2(l)}\right) \& \\ \nexists l_0 \left(x_{l_0 w}^k = 1\right) \& \left(\tau_l^k \leq t_{l_0} \leq t_{l_*}\right) \end{bmatrix};$$

$$Q(k,l) = \exists l_* \exists w \begin{bmatrix} \left(x_{l_* w}^k = 1\right) \& \left(g_1(l) = g_2(l_*)\right) \& \left(\mu_k \leq t_l - \tau_{l_*}^k \leq \eta_{g_1(l)}\right) \& \nexists l_0 \left(x_{l_0 w}^k = 1\right) \& \\ \left(\tau_{l_*}^k \leq t_{l_0} \leq t_l\right) \end{bmatrix}.$$

Общий смысл данного ограничения – каждый ЛА летает по замкнутому маршруту и возвращается в свой базовый аэропорт. В рамках этого маршрута есть посадки, но кратковременные – на дозаправку и посадку и высадку пассажиров, в базовом аэропорте можно стоять сколько угодно;

μ_k – минимальное время, требуемое на подготовку ЛА данного типа к вылету,

η_j – максимальное время, выделяемое в не базовом аэропорту для ЛА (размер слота).

Один рейс в один день выполняет только один ЛА:

$$\forall l \forall w \left[\sum_k x_{lw}^k = 1 \right].$$

Нужно перевести всех (или почти всех) пассажиров:

$$\sum_{j_1=1}^{m} \sum_{j_2=1}^{m} \max \left(1000000 \left[a_{j_1 j_2} - \sum_{l \in \{l: g_1(l) = j_1 \& g_2(l) = j_2\}} \sum_k \sum_w v_{f(k)} x_{lw}^k \right], 0 \right).$$

Требуется минимизировать парк ЛА:

$$\sum_k \tilde{n}_{f(k)} \operatorname{sgn} \left(\sum_w \sum_l x_{lw}^k \right)$$

и минимизировать затраты на конкретные перелеты:

$$\sum_k \sum_w \sum_l \left(q_{f(k)} d_{g_1(l) g_2(l)}^k + w_{f(k)} \right) x_{lw}^k.$$

Итоговая целевая функция:

$$\min_{\substack{x_{lw}^k,\\ t_l,\\ y_k}} \left\{ \begin{array}{l} \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\max\left(1000000\left[a_{j_1 j_2} - \sum_{l\in\{l:g_1(l)=j_1 \& g_2(l)=j_2\}}\sum_k\sum_w v_{f(k)}x_{lw}^k\right],0\right)+ \\ \sum_k \tilde{n}_{f(k)}\operatorname{sgn}\left(\sum_w\sum_l x_{lw}^k\right)+ \\ \sum_k\sum_w\sum_l\left(q_{f(k)}d_{g_1(l)g_2(l)}^k + w_{f(k)}\right)x_{lw}^k \end{array} \right\}.$$

Эту проблему можно решить, и как классическую оптимизационную задачу. Однако подход DCOP обеспечивает масштабируемое решение, которое можно реализовать большом вычислительном кластере, обеспечив, таким образом, существенный рост производительности.

Определение сети ограничений

Ключевым элементом для распределенной обработки ограничений является концепция сети ограничений. Приведем стандартные формальные определения, относящиеся к сетям ограничений, а затем рассмотрим парадигму распределенной обработки ограничений.

СетьограниченийNформальноопределяетсякаккортеж $\langle X, D, C\rangle$,где $X=\{x_1,x_2,...,x_n\}$ – набор дискретных переменных, $D=\{D_1,D_2,...,D_n\}$ – это множество доменов переменных, которые перечисляют все возможные значения соответствующих переменных, а $C=\{C_1,C_2,...,C_n\}$ представляет собой набор ограничений. Ограничение C_i может быть двух типов: жесткое или мягкое.

Жесткое ограничение C_i^h – это отношение R_i, определенное для подмножества переменных $S_i \subseteq X$. Переменные в S_i являются областью действия ограничения, а отношение R_i перечисляет все допустимые совместные комбинации значений переменных в области действия ограничения. Следовательно, R_i является подмножеством декартового произведения доменов переменных, которые находятся в пределах ограничения: $R_i \subseteq D_{i_1}\times D_{i_2}\times ...\times D_{i_r}$, где $r=|S_i|$ – это арность отношения. Мягкое ограничение C_i^s – это функция F_i, определенная для подмножества переменных $S_i \subseteq X$, которые составляют область определения функции. Каждая функция F_i отображает каждую комбинацию значений всех переменных в области видимости на вещественное значение, то есть $F_i : D_{i_1}\times D_{i_2}\times ...\times D_{i_r} \Rightarrow \mathbb{R}$, где $r=|S_i|$ арность функции.

Ограничения могут быть определены для любого подмножества переменных, однако наиболее наглядно работа в сетях ограничений (как алгоритмах решения, так и теоретическом анализе) показывается на бинарных сетях ограничений, где каждое ограничение (мягкое или жесткое) определяется двумя переменными.

Если набор ограничений включает только жесткие ограничения, то проблема называется проблемой удовлетворения ограничений (constraint satisfaction problem – CSP). В этом случае, ищутся значения для всех переменных в сети, которые удовлетворяют всем ограничениям. Значения переменных удовлетворяют ограничению, если кортеж значений переменных принадлежит отношению ограничения, если $(a_{i_1},a_{i_2},...,a_{i_r})\in R_i$, где $a_j \in D_j$ и $S_i=\{x_{i_1},x_{i_2},...,x_{i_r}\}$, то такое назначение относится к решению для сети.

Если набор ограничений включает мягкие ограничения, то мы сталкиваемся с проблемой условной оптимизации (constraint optimization problem – COP) и наша цель – найти оптимальное решение. Это, такая комбинация значений всех переменных, которая удовлетворяет всем ограничениям, и которая максимизирует глобальную функцию $F(\overline{a})$. Глобальная функция $F(\overline{a})$ является суммой функций, представляющих мягкие ограничения: $F(\overline{a}) = \sum F_i(\overline{a}_i)$, где $\overline{a} = (a_1, a_2, ..., a_n)$, где $a_j \in D_j$, и $\overline{a}_i$ – это подмножество $\overline{a}$ для S_i. COP – может быть задачей максимизации или минимизации. Без ограничения общности далее сосредоточимся на задачах максимизации, поэтому наша цель – найти такое $\overline{a}^*$, которое удовлетворяет всем жестким ограничениям и такое, что:

$$\overline{a}^* = \arg\max_{\overline{a}} \sum_i F_i(\overline{a}_i).$$

В целом, каждая задача CSP может также рассматриваться как задача COP, где мы стремимся найти назначение, которое нарушает наименьшее количество ограничений. Это особенно полезно для задач с большим количеством ограничений, где решение для CSP может не существовать. В частности, мы можем назначить постоянную фиксированную стоимость каждому нарушенному ограничению и искать назначение, которое минимизирует сумму затрат. Такая задача называется Max-CSP.

Можно реализовать жесткие ограничения с помощью мягких ограничений. Для этого вводится существенный штраф за комбинации значений переменных, которые не допустимы.

Например, предположим, без ограничения общности, что решается задача максимизации, и пусть R_i – отношение, соответствующее строгому ограничению C_i. Мы можем построить функцию $F_i(\overline{a}_i) = -\infty$, если $\overline{a}_i \notin R_i$ и $F_i(\overline{a}_i) = 0$, в остальных случаях. Следует отметить, что это может привести к ухудшению процедуры поиска решения, так как явные жесткие ограничения могут быть использованы для сокращения пространства поиска решения [4].

Метод решения

Распределенное решение предполагает использование набора агентов, которые контролируют переменные и взаимодействуют, чтобы найти решение для сети ограничений. Как было сказано выше, это могут быть задачи CSP или COP, которые решаются соответствующими распределенными методами: распределенный CSP (distributed CSP – DCSP) и распределенный COP (distributed COP – DCOP). Парадигма DCSP была первоначально предложена для решения проблем координации в среде с несколькими агентами [5], однако в последние годы платформе DCOP уделялось больше внимания, поскольку она имеет больше сценариев практического применения и CSP можно свести к COP.

DCOP представлен сетью $N = \langle X, D, C \rangle$, содержащей мягкие ограничения, плюс набор агентов $A = \{A_1, A_2, ..., A_k\}$. Поиск оптимального решения DCOP – это NP-трудная проблема. Поэтому эмпирическая оценка методов решения DCOP является решающим моментом для оценки их возможного практического применения.

Учитывая предыдущее описание DCOP, рассмотрим точные методы решения, то есть те, которые всегда находят решение, которое соответствует наилучшему значению целевой функции (глобальный оптимум). Эти методы особенно интересны и изящны с теоретической точки зрения, но, поскольку мы имеем дело с проблемой NP-полноты, они также демонстрируют экспоненциально увеличивающиеся издержки координации (либо в размере, либо количестве сообщений, которыми обмениваются, либо в вычислениях, проводимых каждым агентом), количество агентов в системе увеличивается.

В целом, подходы можно разделить на два класса: те, которые основаны на поиске [6–10], и те, которые используют динамическое программирование [11]. Кроме того, подходы, основанные на поиске, делятся на синхронные, такие как SyncBB[10] и AND/OR поиск [9], и асинхронные, такие как ADOPT[6], NCBB[7] и AFB[8]. В модели синхронного выполнения агенты ждут сообщений от других агентов, прежде чем вычислить и отправить новые сообщения самим. Напротив, в асинхронной модели агенты выполняют вычисления и отправляют сообщения, не ожидая сообщений от своих соседей. Асинхронная операция желательна в мультиагентном подходе, поскольку она позволяет агентам принимать решения, не дожидаясь, пока другие агенты завершат свои вычисления, полностью используя параллельные вычисления. С другой стороны, синхронная модель гарантирует, что агенты всегда обладают самой актуальной информацией перед выполнением вычислений, таким образом, сводя к минимуму избыточность как в вычислениях, так и в коммуникации.

Все вышеперечисленные методы полностью децентрализованы, в том смысле, что каждый агент имеет полный контроль над своими переменными и знает только про релевантные ограничения. Тем не менее, централизация части проблемы иногда может уменьшить усилия, необходимые для поиска оптимального в глобальном масштабе решения. Данная концепция, лежит в основе подхода оптимального асинхронного частичного наложения (Optimal Asynchronous Partial Overlay – optAPO) [12]. Алгоритм optAPO стремится обнаружить части задачи, которые особенно трудно решить децентрализованным способом (части, которые сильно взаимосвязаны). Далее он объединяет их в подзадачи, которые делегируются агентам-посредникам, действующим как централизованные решатели. Практика показала, что optAPO последовательно снижает накладные расходы на коммуникации по сравнению с другими децентрализованными методами, такими как ADOPT. Однако очень трудно предугадать, какая часть задачи будет решаться централизовано, и, следовательно, трудно предсказать затраты вычислительных ресурсов, которые потратят агенты-посредники.

В работе проведены вычислительные эксперименты на децентрализованном подходе ADOPT. Для решения поставленной задачи использовался пакет pyDCOP – это решатель DCOP написанный на языке Python[13]. Он обладает следующими особенностями:

- предоставляет реализации многих классических алгоритмов DCOP;
- позволяет легко реализовать собственные алгоритмы DCOP, предоставляя всю необходимую инфраструктуру: агенты, систему обмена сообщениями, сбор метрик;
- упрощает проведение распределенных экспериментов, так как агенты могут работать на одном и том же компьютере или на разных ЭВМ;
- обеспечивает мультиплатформенность, так как может работать на Windows, Mac и Linux;
- подходит для использования в интернете вещей (IoT) и может запускать агенты на одноплатных компьютерах, таких как Raspberry Pi.

Поставленная задача формирования расписания была успешно решена на языке Python и размещена на портале веб-сервисов поддержки принятия решений ws-dss.com (см. рис. 1).

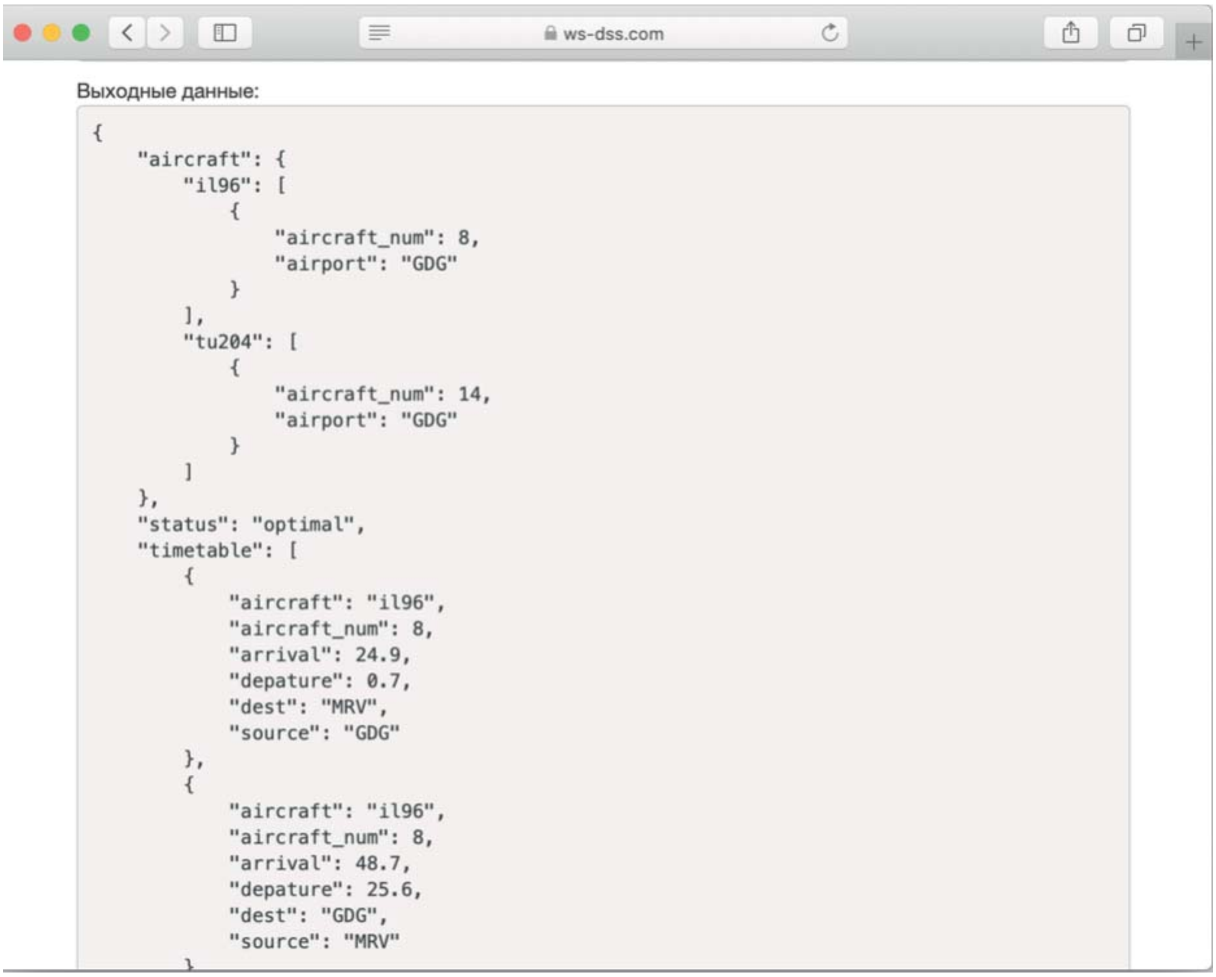

Рис. 1. Решение задачи на ws-dss.com

Несколько работ основаны на ADOPT, пытаются сократить время вычислений. Например, в работе [14] предложен метод BnB-ADOPT, который является расширением ADOPT, он последовательно сокращает время вычислений, используя различные стратегии поиска: поиск в глубину и метод ветвей и границ. В работе [15] предложено использование методов предварительной обработки для поиска ADOPT и показано, что это может привести к существенному увеличению производительности.

Заключение

ADOPT – эффективный мультиагентный метод, который работает в асинхронном режиме. Использование памяти каждым агентом является полиномиальным по количеству переменных, что является существенным преимуществом данного подхода по сравнению с динамическим программированием. Кроме того, все сообщения имеют фиксированный размер. Однако количество сообщений, которыми должны обмениваться агенты, в худшем случае экспоненциально по количеству переменных. Это влияет на время, необходимое для поиска оптимального решения. В частности, число циклов синхронизации сообщений определяется числом всех агентов, получивших входящие сообщения и отправивших исходящие сообщения, и является экспоненциальным. Такие экспоненциальные элементы неизбежны при поиске точного оптимального решения и могут серьезно ограничивать размерность решаемых задач.

MAS позволяет минимизировать количество информационных агентов, которые должны раскрывать информацию друг другу (таким образом, повышается уровень конфиденциальности). Это связано с тем, что в DCOP агентам необходимо знать только об ограничениях, в которые они вовлечены.

В целом, структура DCOP и алгоритмы, разрабатываемые для решения таких проблем, представляют собой активную область исследований в сообществе MAS, которая все чаще применяется в реальных условиях.

Финансирование

Работа выполнена при поддержке гранта РФФИ проект № 18–00–00012 (18–00–00011) КОМФИ.

Литература

1. *A. Farinelli, M. Vinyals, A. Rogers, and N. Jennings*. “Distributed Constraint Handling and Optimization”, in G. Weiss (ed.), “Multiagent Systems” (second edition), MIT Press, p. 547–584, 2013.
2. *Пантелеев А.В., Метлицкая Д.В., Алешина Е.А.* Методы глобальной оптимизации. Метаэвристические стратегии и алгоритмы. М.: Вузовская книга, 2013. 244 с.
3. *Сивакова Т.В., Судаков В.А.* Метод нечетких областей предпочтении для оценки эффективности инноваций // XXVIII Международная научно-техническая конференция «Современные технологии в задачах управления, автоматики и обработки информации». Алушта, 14–20 сентября 2019 г. Сборник трудов. М.: Изд.-во Национальный исследовательский ядерный университет «МИФИ», 2019. С. 81–82.
4. *R. Dechter.* Constraint Processing. Morgan Kaufmann, 2003.
5. *Makoto Yokoo.* Distributed constraint satisfaction: Foundations of cooperation in multi-agent systems. Springer-Verlag, 2001.
6. *P.J. Modi, W. Shen, M. Tambe, and M. Yokoo.* ADOPT: Asynchronous distributed constraint optimization with quality guarantees. Artificial Intelligence Journal, (161):149–180, 2005.
7. *A. Chechetka and K. Sycara.* No-commitment branch and bound search for dis- tributed constraint optimization. In Proceedings of Fifth International Joint Confer- ence on Autonomous Agents and Multiagent Systems, pages 1427–1429, 2006.
8. *Gershman, A. Meisels, and R. Zivan.* Asynchronous forward bounding for dis- tributed COPs. Journal Artificial Intelligence Research, 34:61–88, 2009.
9. *R. Dechter and R. Mateescu.* And/or search spaces for graphical models. Artificial Intelligence, 171:73–106, 2007.
10. *Katsutoshi Hirayama and Makoto Yokoo.* Distributed partial constraint satisfaction problem. In Principles and Practice of Constraint Programming, pages 222–236, 1997.
11. *A. Petcu and B. Faltings.* DPOP: A scalable method for multiagent constraint optimization. In Proceedings of the Nineteenth International Joint Conference on Arti- ficial Intelligence, pages 266–271, 2005.
12. *R.Maillerand, V.Lesser.* Solving distributed constraint optimization problems using cooperative mediation. In Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems, pages 438–445, 2004.
13. Library for research on Distributed Constraints Optimization Problems. URL: https://github.com/Orange-OpenSource/pyDcop (датаобращения: 26.10.2019)

14. *W. Yeoh, A. Felner, and S. Koenig.* BnB-ADOPT: An asynchronous branch-and- bound DCOP algorithm. In Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, pages 591–598, 2008.
15. *S.M. Ali, S. Koenig, and M. Tambe.* Preprocessing techniques for accelerating the DCOP algorithm ADOPT. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pages 1041–1048, 2005.

Multi-Agent Modeling in Schedule Problems

Sivakova T.V.*
Keldysh Institute of Applied Mathematics, Moscow, Russia
Plekhanov Russian University of Economics, Moscow, Russia
sivakova15@mail.ru

Sudakov V.A.**
Moscow Aviation Institute, Moscow, Russia
Keldysh Institute of Applied Mathematics, Moscow, Russia
sudakov@ws-dss.com

The article explores the use of multi-agent technologies for solving optimization problems. It is shown how multi-agent systems allow working with restrictions in a distributed computing environment. The task of scheduling is formalized. Software was developed and computational experiments were carried out, which showed the effectiveness of the proposed approach.

Keywords: multiagent systems, agent preferences, optimization, distributed systems.

Funding
This work was supported by grant RFBR No 18–00–00012 (18–00–00011) KOMFI.

References

1. A. Farinelli, M. Vinyals, A. Rogers, and N. Jennings. "Distributed Constraint Handling and Optimization", in G. Weiss (ed.), "Multiagent Systems" (second edition), MIT Press, p. 547–584, 2013.
2. Panteleev A.V., Metlitskaya D.V., Aleshina E.A. Metody global'noi optimizatsii. Metaevristicheskie strategii i algoritmy [Global optimization methods, Metaheuristic strategies and algorithms]. Moscow, Vuzovskayakniga, 2013. 244 p. (in Russian)
3. Sivakova T.V., Sudakov V.A. Metod nechetkih oblastej predpochtenii dlya ocenki effektivnosti innovacij [Fuzzy preference method for evaluating innovation performance] // XXVIII Mezhdunarodnaya nauchno-tekhnicheskaya konferenciya «Sovremennye tekhnologii v zadachah upravleniya, avtomatiki i obrabotki informacii». Alushta, 14–20 sentyabrya 2019 g. Sbornik trudov. M.: Izd.-vo Nacional'nyj issledovatel'skij yadernyj universitet "MIFI", 2019. 81–82. (in Russian)

For citation:
Sivakova T.V., Sudakov V.A. Multi-Agent Modeling in Schedule Problems. *Modelirovanie i analiz dannykh = Modelling and Data Analysis,* 2019. Vol. 09, no. 4, pp. 100–111. doi: 10.17759/mda.2019090408 (In Russ., abstr. in Engl.)

****Sivakova Tatyana Vladimirovna,*** Researcher, Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), researcher, Plekhanov Russian University of Economics, Moscow, Russia. E-mail: sivakova15@mail.ru

*****Sudakov Vladimir Anatolievich,*** Doctor of Technical Sciences, Professor, Moscow Aviation Institute (National Research University), Leading Researcher, Keldysh Institute of Applied Mathematics (Russian Academy of Sciences), Moscow, Russia. E-mail: sudakov@ws-dss.com

4. R. Dechter. Constraint Processing. Morgan Kaufmann, 2003.
5. Makoto Yokoo. Distributed constraint satisfaction: Foundations of cooperation in multi-agent systems. Springer-Verlag, 2001.
6. P.J. Modi, W. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. Artificial Intelligence Journal, (161):149–180, 2005.
7. A. Chechetka and K. Sycara. No-commitment branch and bound search for distributed constraint optimization. In Proceedings of Fifth International Joint Confer- ence on Autonomous Agents and Multiagent Systems, pages 1427–1429, 2006.
8. Gershman, A. Meisels, and R. Zivan. Asynchronous forward bounding for dis- tributed COPs. Journal Artificial Intelligence Research, 34:61–88, 2009.
9. R. Dechter and R. Mateescu. And/or search spaces for graphical models. Artificial Intelligence, 171:73–106, 2007.
10. Katsutoshi Hirayama and Makoto Yokoo. Distributed partial constraint satisfaction problem. In Principles and Practice of Constraint Programming, pages 222–236, 1997.
11. A. Petcu and B. Faltings. DPOP: A scalable method for multiagent constraint optimization. In Proceedings of the Nineteenth International Joint Conference on Arti- ficial Intelligence, pages 266–271, 2005.
12. R.Maillerand, V.Lesser. Solving distributed constraint optimization problems using cooperative mediation. In Proceedings of Third International Joint Conference on Autonomous Agents and MultiAgent Systems, pages 438–445, 2004.
13. Library for research on Distributed Constraints Optimization Problems. URL: https://github.com/Orange-OpenSource/pyDcop (26.10.2019)
14. W. Yeoh, A. Felner, and S. Koenig. BnB-ADOPT: An asynchronous branch-and- bound DCOP algorithm. In Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems, pages 591–598, 2008.
15. S.M. Ali, S. Koenig, and M. Tambe. Preprocessing techniques for accelerating the DCOP algorithm ADOPT. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, pages 1041–1048, 2005.