

Роевой алгоритм планирования работы многопроцессорных вычислительных систем

Б.К. Лебедев, О.Б. Лебедев, Е.О. Лебедева

Южный федеральный университет, Таганрог

Аннотация. В работе рассматривается задача составления плана выполнения комплекса программ, в многопроцессорных вычислительных системах (МВС). МВС состоит из нескольких параллельно работающих процессоров. На вход МВС поступает множество независимых потоков заявок (программ), которые необходимо распределить между процессорами. Вычислительная система может состоять как из идентичных, так и из различных по производительности процессоров. Учитывается время переключения между различными классами заявок, поступающих на процессор. Решение задачи планирования представляется как задание распределения заявок по процессорам, и определение очереди заявок на обслуживание процессором. Оптимизация при планировании в случае многоуровневой очереди заключается в минимизации времени выполнения всех заявок. В основу работы представленного алгоритма положены механизмы адаптивного поведения муравьиной колонии. Временная сложность этого алгоритма зависит от времени жизни колонии (число итераций), количества исполнителей и числа работ.

Ключевые слова: многопроцессорная система, планирование, многоуровневая очередь, распределительная задача, оптимизация, муравьиный алгоритм.

Введение

Одна из наиболее распространенных систем массового обслуживания (СМО), осуществляет обработку заявок, рассортированных по разным группам в многоуровневые очереди [1]. Подобные задачи возникают во многих областях, требующих организации эффективного планирования заданий, работ, требований. Имеется некоторое число *заявок* и некоторое число *обслуживающих приборов*. Обработка любой заявки может быть выполнена любым обслуживающим прибором, но с неодинаковыми затратами. Нужно распределить заявки так, чтобы обработать их обслуживающими приборами с наименьшими затратами. В качестве примера можно привести задачу составления плана выполнения комплекса программ, в многопроцессорных вычислительных системах [2]. Вычислительная система состоит из набора параллельно работающих независимых

процессоров, что позволяет выполнять сразу несколько программ одновременно. Вычислительная система может состоять как из идентичных, так и из различных по производительности процессоров.

Имеется в виду, что выполняемая программа представляется в виде неделимого целого, в связи, с чем прерывание ее работы и передача на исполнение другому процессору, в общем случае, запрещены. Каждая программа может выполняться на любом из процессоров, при этом время исполнения программы зависит от ее сложности и от быстродействия процессора. Длительность выполнения программы считается заранее известной величиной. Если для исполнения программы требуются данные получаемые в результате работы другой программы, то на множестве программ задается отношение порядка следования. В противном случае, программы считаются независимыми. Таким образом, требуется наилучшим по быстродействию или равномерности загрузки образом спланировать выполнение комплекса программ на заданном множестве процессоров, а в случае наличия связи между программами, также определить последовательность их исполнения на каждом процессоре. Рассмотренная задача планирования вычислительных процессов в многопроцессорных и многомашинных комплексах является минимаксной распределительной задачей (P3) теории расписаний.

Российские и зарубежные ученые разработали большое число алгоритмов и методов решения распределительных задач различающихся как эффективностью, так и областью применения. Поскольку P3 принадлежат к классу NP-полных задач, разработка эффективных алгоритмов их решения по прежнему остается актуальной проблемой теории расписаний. Разработанные методы решения распределительной задачи принято разделять на два больших класса: точные и приближенные. Среди существующих известных и эффективных точных методов решения P3

наибольшее распространение получили алгоритмы, построенные по схеме метода ветвей и границ (МВГ) [3]. Существенным недостатком метода ветвей и границ является экспоненциальное увеличение сложности решения относительно размерности РЗ. В ряде работ [4] предлагаются подходы, позволяющие сократить комбинационную сложность алгоритмов, построенных по схеме метода ветвей и границ [2]. Особенностью метода ветвей и границ является то, что всегда существует вероятность полного перебора и необходимы достаточно большие временные затраты, особенно при решении задач повышенной размерности.

В связи с этим широкое распространение получили приближенные методы, ориентированные на получение некоторого приемлемого или допустимого решения [5;6]. Достоинство таких методов состоит в высокой скорости решения, характеризующиеся полиномиальной или, даже, линейной зависимостью от размерности задачи. К числу недостатков следует отнести невысокую точность методов данного класса. К числу наиболее распространенных приближенных методов относятся так называемые списочные и эвристические методы. Обладая в основном, линейной сложностью решения относительно размерности задачи, списочные методы крайне эффективны с точки зрения времени решения РЗ. С другой стороны, отклонение получаемых решений от оптимального значения может достигать 30% , что часто не удовлетворяет предъявляемым требованиям к решениям. Высокая трудоемкость точных методов и низкая эффективность приближенных списочных методов резко снизили их практическое применение. В связи с этим большая часть разработок была направлена на применение эвристических подходов. Анализ существующих алгоритмов (последовательных, итерационных и т.д.), приведенный в [5-7] показывает, что для разработки более эффективного алгоритма, который отвечает предъявляемым современным требованиям, необходимы новые технологии и

подходы. В последние годы интенсивно разрабатывается научное направление, объединяющее математические методы, в которых заложены принципы природных механизмов принятия решений [8,9]. Одним из таких направлений являются мультиагентные методы интеллектуальной оптимизации, базирующиеся на моделировании коллективного интеллекта [10-13].

Такие методы являются итеративными, эвристическими методами случайного поиска. Особенно наблюдается стремительный рост интереса к разработке алгоритмов, инспирированных природными системами. Среди них активно развиваются методы роевого интеллекта (Swarm Intelligence) [8-10], в которых совокупность простых агентов конструирует стратегию своего поведения без наличия глобального управления. Одним из новейших мультиагентных методов интеллектуальной оптимизации является метод муравьиной колонии [9]. В работе излагается метод планирования комплекса программ в многопроцессорных вычислительных системах, базирующийся на моделировании адаптивного поведения муравьиной колонии.

1. Постановка задачи оптимизации обслуживания многоочередных структур данных

Пусть МВС состоит из n идентичных, параллельно работающих процессоров $E = \{e_i | i = 1, 2, \dots, n\}$. На вход МВС поступает множество m независимых программ $W = \{w_j | j = 1, 2, \dots, m\}$, которые требуется распределить между процессорами. Если время r_j непрерывного использования процессора для выполнения каждой программы w_j , одинаково для всех процессоров, то множеству W сопоставляется множество $R = \{r_j | j = 1, 2, \dots, m\}$. l -м решением задачи планирования в этом случае является множество $V^l = \{W_i^l | i = 1, 2, \dots, n\}$, где W_i^l подмножество программ, выполняемых на процессоре e_i :

$$(\forall i, j)[((i \neq j) \& (W_i^l \in V^l) \& (W_j^l \in V^l)) \rightarrow ((W_i^l \cap W_j^l = \emptyset) \& (\cup W_i^l = W))]. \quad (1)$$

Запланированная вариантом решения V^l загрузка программами каждого процессора e_i оценивается ресурсом:

$$R_i^l = \sum_k r_k, \text{ где } r_k | w_k \in W_i^l. \quad (2)$$

Если производительность процессоров различна, то задается время r_{ij} непрерывного использования процессора e_i для выполнения программы w_j . Таким образом, множествам E и W сопоставляется множество $R = \{r_{ij} | i=1,2,\dots,n; j=1,2,\dots,m\}$. В этом случае формула (2) примет вид:

$$R_i^l = \sum_k r_{ik}, \text{ где } k | w_k \in W_i^l. \quad (3)$$

Представим некоторую очередь программ, поступающих на процессор e_i в виде списка $P_i^l = p_1, p_2, p_3, \dots, p_s, \dots, p_m$, где p_s это индекс программы, у которой порядковый номер в очереди P_i^l равен s .

Загрузка программами каждого процессора, рассчитываемая по формулам (2) или (3), не учитывает время переключения между различными заявками. Отметим, что время переключения для каждой пары соседних заявок в очереди имеет свое значение.

Продолжительность переключения между соседними в очереди заявками, поступающими на обслуживающий прибор e_i , задается матрицей переключений $T_i = ||t_{k,s}||_{m \times m}$.

Таблица № 1

Матрица переключений T_i

*	t_{12}	t_{13}	t_{14}	t_{15}
t_{21}	*	t_{23}	t_{24}	t_{25}
t_{31}	t_{32}	*	t_{34}	t_{35}
t_{41}	t_{42}	t_{43}	*	t_{45}
t_{51}	t_{52}	t_{53}	t_{54}	*

$t_{k,s}$ – время переключения между соседними в очереди заявками (w_k, w_s).

Очередь заявок, поступающих на обслуживающий прибор, с учетом времени переключения между различными соседними заявками примет вид:

$$P_i^l = p_1, t_{12}, p_2, t_{23}, p_3, \dots, p_s, t_{s,s+1}, p_{s+1}, \dots, p_{m-1}, t_{m-1,m}, p_m.$$

Здесь $t_{s,s+1}$ - время переключения между соседними в очереди P_i^l программами с порядковыми номерами s и $s+1$.

Общее время T_i^l переключения между заявками, поступающими на вход обслуживающего прибора e_i , определится как сумма переключений:

$$T_i^l = \sum_k t_{k-1,k}, \text{ где } k | t_{k-1,k} \in P_i^l. \quad (4)$$

Таким образом, нагрузка R_i^l заявками процессора e_i с учетом времени переключения между соседними заявками в очереди P_i^l , а также времени r_{ij} непрерывного использования процессора e_i для выполнения каждой программы $w_j \in W_i^l$ определится как:

$$R_i^l = T_i^l + \sum_k r_{ik}, \text{ где } k | w_k \in W_i^l \quad (5)$$

Для обеспечения максимальной эффективности использования существующих ресурсов необходимо упорядочить очередь заявок в каждом списке P_i^l . Будем считать, что $P^l = \{P_i^l | i=1, 2, \dots, n\}$ - множество упорядоченных списков, задающих очередность обработки заявок на обслуживающих приборах.

В этом случае решением Z^l задачи распределения заявок между обслуживающими приборами является совокупность двух множеств V^l и P^l , т.е. $Z^l = (V^l, P^l)$.

В качестве оценки решения Z^l рассматривается величина:

$$F^l = \max_i (R_i^l). \quad (6)$$

Наиболее распространённым критерием оптимизации при планировании в случае многоуровневой очереди является «минимаксный» критерий:

$$F = \min_l F^l = \min_l \max_i (R_i^l). \quad (7)$$

Это связано с тем, что в большинстве реальных задач при планировании чаще всего требует решения задача уменьшения времени выполнения всех заявок. Поскольку, чем быстрее СМО обслуживает поступающие заявки, тем выше ее производительность, а значит, и выше экономическая эффективность предприятия, использующего данную систему.

Для отображения распределения заявок по обслуживающим приборам используется полный двудольный граф $H_{nm}=(E \cup W, U)$, где $E=\{e_i | i=1, 2, \dots, n\}$ - множество вершин (первая доля), соответствующих множеству обслуживающих приборов, а $W=\{w_j | j=1, 2, \dots, m\}$ - множество вершин (вторая доля), соответствующих множеству заявок (поток). U - множество ребер, связывающих вершины множества E с вершинами множества W . Конкретное решение задачи распределения заявок представляется в виде подграфа $H^l=(E \cup W, U^l)$. $H^l \subset H_{nm}$, $U^l \subset U$. Вершина $w_j \in W$ связана с вершиной $e_i \in E$ в том и только в том случае, когда w_j назначена в e_i . Пусть U^l_i множество ребер двудольного графа H^l , связывающих вершину e_i с вершинами множества W^l_i . $\cup U^l_i = U^l$. Для наглядности представления конкретного распределения заявок (рис. 1) множество заданий W сгруппировано в подмножества W^l_i , где W^l_i - подмножество заявок, поступающих на обслуживающий прибор e_i .

Отличительной особенностью представленного двудольного графа $H^l=(E \cup W, U^l)$ является то, что число ребер множества U^l_i равно числу вершин множества W^l_i . Каждое ребро $u_z \in U^l_i$, с одной стороны, инцидентно вершине e_i , а с другой стороны, инцидентно одной и только вершине $w_k \in W^l_i$.

Назовем двудольный граф, представляющий распределение заявок, граф - распределением H^l . Отметим, что в графе - распределении H^l локальная степень любой вершины $w_j \in W$ равна единице, а локальная степень вершины e_i равна мощности множества W^l_i , т.е. $\rho(w_j)=1$, $\rho(e_i)=|W^l_i|$.

Таким образом, поиск распределение заявок сводится к поиску на полном двудольном графе H_{nm} подграфа H^l .

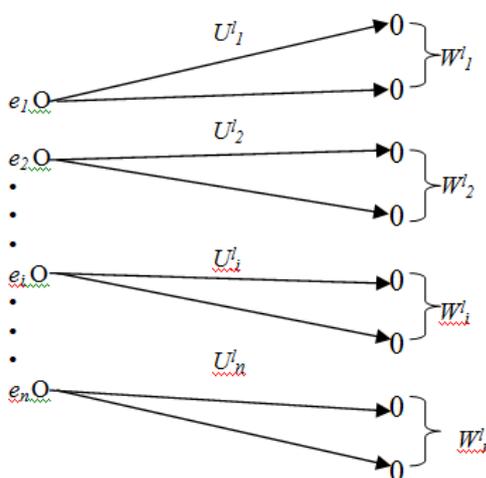


Рис. 1 - Интерпретация распределения программ в MBC

2. Разработка алгоритма планирования на основе моделей адаптивного поведения муравьиной колонии

В основу работы разработанного алгоритма положены механизмы адаптивного поведения муравьиной колонии. Особенности являются наличие непрямого обмена информацией, который используется в муравьиных алгоритмах. Непрямой обмен – стигмержи представляет собой разнесённое во времени взаимодействие, при котором одна особь оставляет результаты своей деятельности в окружающей среде, а другие используют эту информацию позже, выполняя в той же среде аналогичные действия. Такое отложенное взаимодействие, использующее в качестве посредника окружающую среду, происходит через специальное химическое вещество – феромон (pheromone).

Метаэвристика муравьиного алгоритма базируется на комбинации двух технологий поиска. Общая структура строится на базовом методе, в который включается та или иная встроенная процедура. Встроенная процедура – это в большинстве случаев самостоятельный алгоритм решения той же задачи, что

и метаэвристический метод в целом. Базовый метод заключается в реализации адаптивной итерационной процедуры поиска лучшего решения среди решений, формируемых с помощью встроенной процедуры. В качестве встроенной процедуры служит конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения. Конструктивный алгоритм, построенный на модели адаптивного поведения искусственных муравьев, играет ключевую роль [8]. Не менее важное значение имеет модель окружающей среды, используемая в качестве посредника при обмене информацией. Классификация гибридных метаэвристик детально рассмотрена в работе [14-16].

В нашем случае в качестве интерпретации распределения заявок по обслуживающим приборам служит двудольный граф – решения H^l .

Рассмотрим принципы планирования методами муравьиной колонии. Будем считать, что подсчет величины R_i^l производится по формуле (5). Поиск оптимального решения осуществляется по критерию F (см.7).

Работа поисковой процедуры начинается с построения в соответствии со спецификой решаемой задачи графа поиска решений (ГПР). Структура ГПР должна позволять формировать отображение и поиск интерпретации ID^l решения $Z^l=(V^l;P^l)$. ГПР формируется на основе интеграции трех составляющих – полного двудольного графа H_{nm} , полного графа G и звездного графа GZ .

На первом этапе формируется полный двудольный граф $H_{nm}=(E \cup W, UI)$, используемый для построения интерпретации множества V^l . Вторая составляющая графа поиска решений связана с необходимостью упорядочивания заявок, т.е. фактически введения приоритета для заявок каждой группы потоков, обслуживаемых одним прибором. Этот этап связан с диверсификацией графа поиска решений и организацией поиска решений на этом графе.

На втором этапе построения графа поиска решений формируется граф $GH_{nm}=(E \cup W, U1 \cup U2)$ на основе интеграции двух графов: полного двудольного графа $H_{nm}=(E \cup W, U1)$ и полного графа $G=(W, U2)$. Другими словами GH_{nm} состоит из двух подграфов - $H_{nm}=(E \cup W, U1)$ и подграфа $G=(W, U2)$, использующих одно множество вершин W . $|U1|=nm$, $|U2|=m(m-1)$. В графе $GH_{nm}=(E \cup W, U1 \cup U2)$ каждая вершина $e_i \in E$ связана со всеми вершинами множества W . Каждая вершина $w_j \in W$, с одной стороны связана со всеми вершинами множества E , а с другой стороны связана со всеми вершинами множества W . Третий этап диверсификации ГПР связан с добавлением третьей составляющей ГПР - звездного графа $GZ=(W \cup O, U3)$. Используемая в качестве стартовой вершины центральная вершина O звездного графа GZ , связывается множеством ребер $U3$ с каждой $w_j \in W$, $|U3|=m$. Звездный граф GZ предназначен для выбора первой вершины $w_j \in W$, включаемой в состав маршрута M^l . Таким образом, диверсифицированный граф поиска решений имеет вид: $D=H_{nm} \cup G \cup GZ$, $D=(E \cup W \cup O, U1 \cup U2 \cup U3)$.

Отметим, все три составляющие графа D – полный двудольный граф H_{nm} , полный граф G и звездный граф GZ используют одно множество вершин W .

Локальные степени вершин графа D : $\rho(e_i)=m$. $\rho(w_j)=n+m$. $\rho(O)=m$.

На рис. 2 приведен диверсифицированный граф поиска решений D для $|E|=3$, $|W|=5$.

$$U1=\{(e_i, w_j) | i=1, 2, 3; j=1, 2, 3, 4, 5\}.$$

$$U2=\{(w_i, w_j) | i \neq j; i=1, 2, 3, 4, 5; j=1, 2, 3, 4, 5\}.$$

$$U3=\{(O, w_j) | j=1, 2, 3, 4, 5\}.$$

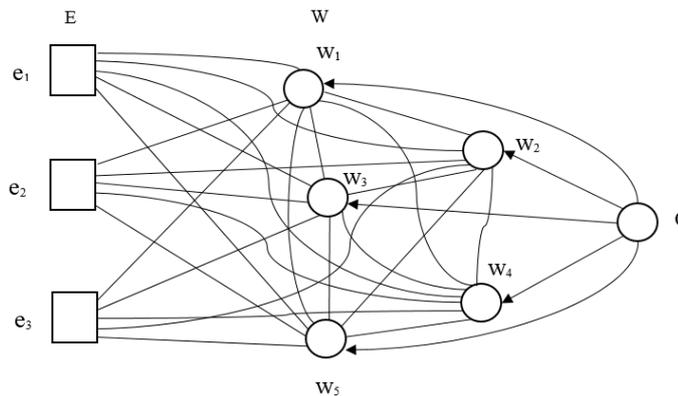


Рис. 2 - Диверсифицированный граф поиска решений

Интерпретация ID^l на ГПР D решения задачи загрузки заявками обслуживающих приборов с учетом времени t_{ks} переключения между различными заявками в очереди P^l_i , а также времени r_{ij} непрерывного использования процессора e_i для выполнения каждой заявки w_j включает две части. Первая часть представляет собой маршрут M^l , построенный агентом, стартовым из вершины O на полном подграфе $G=(W, U_2)$ графа D . Маршрут M^l начинается с вершины O и включает все вершины подграфа G .

Вторая часть представляет собой подграф $H^l=(E \cup W, U_1^l)$ полного двудольного графа $H_{nm}=(E \cup W, U_1)$, $U_1^l \subset U_1$, $|U_1^l|=m$. H^l отражает разбиение множества вершин W на подмножества W^l_i . $ID^l=(M^l, H^l)$.

На основе маршрута M^l и подграфа H^l для каждого обслуживающего прибора e_i определяется множество заявок W^l_i и формируется очередь P^l_i . Заявки каждого подмножества W^l_i поступают на обслуживающий прибор e_i в том порядке, в котором они расположены в маршруте M^l друг относительно друга. Время t_{ks} переключения между соседними заявками (w_k, w_s) в очереди P^l_i определяется из таблицы 1.

Пример. Пусть на графе поиска решений D , представленном на рис. 2, построен маршрут $M^l = o \rightarrow w_1 \rightarrow w_5 \rightarrow w_2 \rightarrow w_3 \rightarrow w_4$ и сформирован граф H^l , задающий разбиение множества заявок W на подмножества: $W^l_i = \{w_1, w_5, w_2\}$;

$W_2^l = \{w_3\}$; $W_3^l = \{w_4\}$. Распределение заявок по обслуживающим приборам, с учетом времени t_{ks} переключения между различными заявками, представлено на рис. 3. На рис. 4 представлено решение $Z^l = (V^l; P^l)$ соответствующее $ID^l = (M^l, H^l)$.

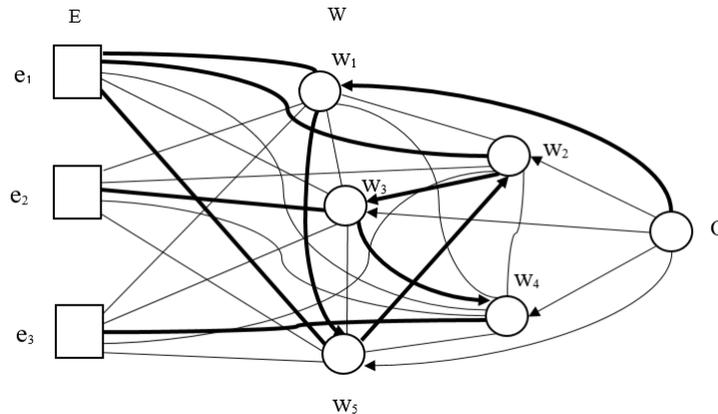


Рис. 3 - Распределение заявок по обслуживающим приборам, с учетом времени

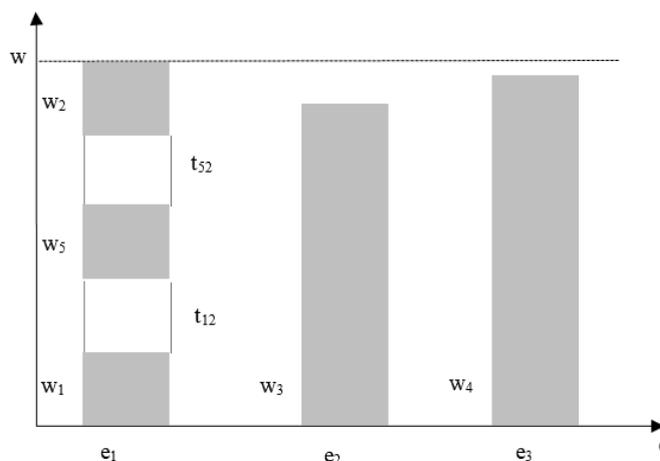


Рис. 4 – Полученное решение

Рассмотрим базовую поисковую процедуру. Поиск решения начинается с задания размера популяции искусственных муравьев. Вершина O в графе D является стартовой для всех муравьев. Моделирование поведения муравьев связано с распределением феромона на ребрах графа D . На начальном этапе на всех ребрах $U = U1 \cup U2 \cup U3$ графа D откладывается одинаковое

(небольшое) количество феромона равно Q/v , где $v=|U|$. Параметр Q задается априори. Будем обозначать граф D после отложения на нем на итерации t феромона, как $D(t)$. После начального отложения - $D(0)$. Процесс поиска решений итерационный. Каждая итерация t включает 3 этапа.

На первом этапе каждой итерации t выполняются процедуры муравья. Каждый агент a_l формирует на ребрах графа $D(t-1)$ интерпретацию решения - свой собственный граф $H^l(t)$ и маршрут $M^l(t)$. На основе интерпретации $ID^l=(M^l(t);H^l(t))$, формируется решение $Z^l(t)=(V^l(t);P^l(t))$ для которого подсчитывается оценка решения $F^l(t)$.

Обозначим как $U^l(t) \subset U$ множество всех рёбер, соответствующих ребрам построенного графа $H^l(t)$ и ребер, входящих в состав маршрута $M^l(t)$. На втором этапе итерации t , каждый муравей a_l откладывает феромон на ребрах множества $U^l(t)$. Количество феромона $\tau^l(t)$, откладываемое муравьем a_l на каждом ребре $u_k \in U^l(t)$, определяется следующим образом:

$$\tau^l(t) = Q / F^l(t),$$

где t -номер итерации, Q – общее количество феромона, откладываемое муравьем на ребрах множества $U^l(t)$, $F^l(t)$ – целевая функция для решения, полученного муравьем a_l на t -ой итерации. Чем меньше $F^l(t)$, тем больше феромона откладывается на ребрах и, следовательно, тем больше вероятность выбора этих ребер при построении интерпретации решения $ID^l=(M^l, H^l)$ на следующей итерации.

Обозначим как $f_k(t)$ суммарное количество феромона, скопившееся на каждом ребре $u_k \in U1 \cup U2 \cup U3$ ГПР, после выполнения второго этапа итерации t .

После того, как каждый агент сформировал решение и отложил феромон, на третьем этапе итерации t происходит общее испарение феромона на каждом ребре u_k ГПР в соответствии с формулой:

$$f_k(t) = f_k(t)(1 - \rho), \quad (8)$$

где ρ – коэффициент обновления.

После выполнения всех действий на итерации t находится агент с лучшим решением, которое запоминается. Далее осуществляется переход на следующую итерацию.

Рассмотрим теперь конструктивный алгоритм построения муравьем интерпретации решения $ID^l = (M^l, H^l)$. Отметим, что маршрут M^l и подграф H^l используют одно множество вершин W . Любой маршрут M^l начинается со стартовой вершины O . Поэтому первой в маршрут M^l включается вершина O .

Начиная с первого шага, маршрут M^l и подграф H^l формируются последовательно и одновременно. Обозначим как W_{end} множество вершин $w_j \in W$, вошедших в формируемый маршрут, а как W_{in} множество свободных вершин $w_j \in W$, еще не вошедших в формируемый маршрут. $W_{end} \cup W_{in} = W$. Пусть вершина $w_\alpha \in W_{end}$, является последней вершиной, включенной на некотором шаге в маршрут M^l , и маршрут еще полностью не сформирован. Параметры W_{end} , w_α , W_{in} являются параметрами состояния процедуры построения маршрута M^l и подграфа H^l .

На первом шаге построения M^l параметры состояния имеют следующие значения: $w_\alpha = O$, $W_{in} = W$, $W_{end} = \emptyset$.

На каждом шаге последовательной процедуры выбирается вершина $w_\beta \in W_{in}$, которая вместе с ребром (w_α, w_β) включается в маршрут M^l и, одновременно для w_β выбирается вершина e_i , что равносильно включению ребра (w_β, e_i) в H^l . Выбор w_β осуществляется следующим образом. Для вершины w_α определяется набор ребер U_{in} , связывающих w_α со всеми вершинами $w_\beta \in W_{in}$. Для каждого ребра $(w_\alpha, w_\beta) \in U_{in}$, связывающего вершину w_α с вершиной $w_\beta \in W_{in}$ определяется параметр $f_{\alpha\beta}$ – суммарный уровень феромона на этом ребре. Вероятность $P_{\alpha\beta}$ включения ребра $(w_\alpha, w_\beta) \in U_{in}$ и вершины w_β , в формируемый маршрут определяется следующим соотношением:

$$P_{\alpha\beta} = f_{\alpha\beta} / \sum_{\beta} (f_{\alpha\beta}). \quad (9)$$

Агент с вероятностью $P_{\alpha\beta}$ выбирает одно из ребер $(w_{\alpha}, w_{\beta}) \in U_{in}$, которое включается в формируемый маршрут.

После этого для выбранной вершины w_{β} определяется набор $UI_{\beta} \subset UI$ ребер, связывающих w_{β} со всеми вершинами $e_i \in E$. Для каждого ребра $(w_{\beta}, e_i) \in UI_{\beta}$, определяется параметр $f_{\beta i}$ – суммарный уровень феромона на этом ребре. Вероятность $P_{\beta i}$ включения ребра $(w_{\beta}, e_i) \in UI_{\beta}$ в формируемый двудольный подграф H^l определяется следующим соотношением:

$$P_{\beta i} = f_{\beta i} / \sum_i (f_{\beta i}). \quad (10)$$

Агент с вероятностью $P_{\beta i}$ выбирает одно из ребер множества UI_{β} , которое включается в формируемый двудольный подграф H^l .

Временная сложность этого алгоритма зависит от времени жизни колонии t (число итераций), количества исполнителей n и числа работ m , и определяется как $O(t \cdot n^2 \cdot m)$.

Ниже приводится описание базового алгоритма поведения муравьиной колонии и конструктивного алгоритма муравья.

Алгоритм поведения популяции агентов (муравьев).

1. Задается: число исполнителей - n ; число работ - m ; начальное количество феромона - Q .

2. Строится граф поиска решений $D = (E \cup W \cup O, UI \cup U2 \cup U3)$.

3. На ребрах ГПР D откладывается начальное количество феромона, равное Q/v , где $v = |U|$.

4. Задается: число итераций - NT ; число муравьев формирующих независимо друг от друга решения на одной итерации - NA .

5. $t = 1$. (t – номер итерации).

6. $l = 1$. (l – номер агента).

7. (*Алгоритм муравья*). Муравей a_l строит на ГПР D интерпретацию решения $ID^l = (M^l, H^l)$.

8. На основе интерпретации решения $ID^l = (M^l; H^l)$ формируется решение $Z^l = (V^l; P^l)$ для которого рассчитывается оценка $F^l(t)$.

9. Если $l < NA$, то $l = l + 1$ и переход к пункту 7, иначе переход к п. 10.

10. $l = 1$.

11. Муравей a_l откладывает на ребрах множества $U^l(t) \subset U$, соответствующих ребрам построенного графа $H^l(t)$ и ребрах, входящих в состав маршрута $M^l(t)$ феромон в количестве $\tau^l(t) = Q / F^l(t)$.

12. Если $l < NA$, то $l = l + 1$ и переход к п. 11, иначе переход к п. 13.

13. На третьем этапе итерации t происходит общее испарение феромона на всех ребрах ГПР D в соответствии с формулой: $f_{ij} = f_{ij}(1 - \rho)$, где ρ – коэффициент обновления.

14. Находится агент с лучшим решением $F^*(t)$, полученным после выполнения t итераций, которое запоминается.

15. Если $t < NT$, то $t = t + 1$ и переход к п. 6, иначе переход к п. 16.

16. Конец работы алгоритма.

Рассмотрим теперь конструктивный алгоритм построения муравьем интерпретации решения $ID^l = (M^l; H^l)$.

Алгоритм муравья.

1. Агент a_l помещается в стартовую вершину O . В маршрут M^l включается вершина O .

2. Формируются параметры состояния: $w_\alpha = O$, $W_{in} = W$, $W_{end} = \emptyset$.

$U1^l = \emptyset$. $H^l = (E \cup W, \emptyset)$

3. Для вершины w_α определяется набор ребер $U2_{in}$, связывающих w_α со всеми вершинами $w_\beta \in W_{in}$.

4. Для каждого ребра $(w_\alpha, w_\beta) \in U2_{in}$, связывающего вершину w_α с вершиной $w_\beta \in W_{in}$ определяется параметр $f_{\alpha\beta}$ – суммарный уровень феромона на этом ребре.

5. Рассчитывается вероятность $P_{\alpha\beta}$ включения ребра $(w_\alpha, w_\beta) \in U_{2in}$ и вершины w_β , в формируемый маршрут M^l , $P_{\alpha\beta} = f_{\alpha\beta} / \sum_{\beta} (f_{\alpha\beta})$.

6. Агент с вероятностью $P_{\alpha\beta}$ выбирает одно из ребер $(w_\alpha, w_\beta) \in U_{2in}$ и вершину w_β .

7. Вершина w_β включается в маршрут M^l .

8. Обновляются значения параметров состояния: $W_{end} = W_{end} \cup w_\beta$, $W_{in} = W_{in} \setminus w_\beta$, $w_\alpha = w_\beta$.

9. Для выбранной вершины w_β определяется набор $U_{1\beta} \subset U_1$ ребер, связывающих w_β со всеми вершинами $e_i \in E$.

10. Для каждого ребра $(w_\beta, e_i) \in U_{1\beta}$, определяется параметр $f_{\beta i}$ – суммарный уровень феромона на этом ребре.

11. Вероятность $P_{\beta i}$ включения ребра $(w_\beta, e_i) \in U_{1\beta}$ в формируемый двудольный подграф H^l определяется следующим соотношением $P_{\beta i} = f_{\beta i} / \sum_i (f_{\beta i})$.

12. Агент с вероятностью $P_{\beta i}$ выбирает одно из ребер $(w_\alpha, w_\beta) \in U_{1\beta}$, которое включается в граф H^l . $U_1^l = U_1^l \cup (w_\alpha, w_\beta)$.

13. Если $W_{in} = \emptyset$, а $W_{end} = W$, то переход к п. 14, иначе переход к п. 3.

14. Интерпретация решения $ID^l = (M^l; H^l)$ полностью сформирована.

Конец работы алгоритма муравья.

3. Экспериментальные исследования

Алгоритм распределения программ на многопроцессорных вычислительных системах был запрограммирован на языке C++ на платформе Windows. При этом все исследования проводились на компьютере типа Intel® Core™ i5 CPU 3.33 GHz и ОЗУ размером 4 Гб. При выполнении экспериментов необходимо было исследовать эффективность и качество предложенных механизмов муравьиной колонии для решения задачи



планирования многоуровневой очереди. Для этого была применена процедура формирования контрольных тестов с известным оптимумом.

На первом этапе проводилось исследование влияния управляющих операторов, таких как: размер популяции муравьев, количество итераций, параметров, управляющих отложением и испарением феромона.

Для получения достоверных выводов был проведен не один, а серия опытов-экспериментов. Временная сложность этого алгоритма зависит от времени жизни колонии t (число итераций), количества обслуживающих приборов n и числа заявок m , и определяется как $O(t \cdot n^2 \cdot m)$. Тестовые испытания показали, что в 98% случаев пространство синтезированных решений включает глобальное оптимальное решение.

Исследование сходимости алгоритма выполнялось следующим образом. Запоминался номер итерации, после которой не наблюдалось улучшения оценки-критерия, для каждого эксперимента. В каждой серии из 50 испытаний определялись минимальный и максимальный номера итерации. Кроме этого рассчитывалось среднее значение количества итераций, после которого не наблюдалось улучшения оценки-критерия. Для каждой серии испытаний определялось лучшее решение, которое фактически являлось оптимальным. В результате экспериментов установлено, что при объеме популяции $M=90$ алгоритм сходится в среднем на 120 итерации.

Сравнение значений критерия, полученных муравьиным алгоритмом на тестовых примерах, с известным оптимумом показало, что у 90% примеров полученное решение было оптимальным, у 15% примеров решения были на 3% хуже оптимального, а у 5% примеров решения были хуже не более, чем на 2%. По сравнению с существующими алгоритмами [3-10] достигнуто улучшение результатов на 2-3%.

Заключение

Несмотря на достаточно большое число разработанных моделей, и алгоритмов оптимизации планирования и диспетчеризации многопроцессорных вычислительных систем, исследователи часто сталкиваются с рядом проблем, к числу которых относятся трудность обоснования качества результатов анализа, учитывающего специфику задачи. Отсюда можно сделать вывод о целесообразности дальнейшего развития методов планирования и диспетчеризации МВС, позволяющих решать указанные проблемы. К числу перспективных тенденций относятся стохастические поисковые алгоритмы оптимизации, которые в разных публикациях называют поведенческими, метаэвристическими, вдохновленными (инспирированными) природой, роевыми, многоагентными, популяционными и т.д. В работе рассматриваются новые принципы решения задачи планирования работы МВС на основе моделей адаптивного поведения муравьиной колонии. Разработаны модифицированные механизмы поведения муравьев и структура пространства решений, в рамках которого организован поисковый процесс, базирующийся на моделировании адаптивного поведения роя муравьев.

Ключевая проблема, которая была решена в данной работе, связана с разработкой композитной структуры пространства решений, позволяющей отображать и осуществлять поиск интерпретаций задачи. Пространство формируется на основе интеграции трех составляющих – полного двудольного графа H_{nm} , полного графа G и звездного графа GZ . Это позволило формировать гибридную интерпретацию решения задачи загрузки заявками обслуживаемых приборов с учетом времени переключения между различными заявками в очереди, а также времени непрерывного использования процессора для выполнения каждой заявки. Поиск решений в соответствии с метаэвристикой муравьиной колонии основывается на комбинации двух алгоритмов: базовый алгоритм реализует итерационную

процедуру поиска лучшего решения и конструктивный алгоритм построения муравьем некоторой конкретной интерпретации решения. Такой подход является источником усовершенствования каждой из процедур в отдельности, что ускорит процесс достижения целевого состояния.

Работа выполнена при финансовой поддержке РФФИ грант №17-07-00997.

Литература

1. Лазарев А.А., Гафаров Е.Р. Теория расписаний. Задачи и алгоритмы. – М.: МГУ им. М.В. Ломоносова, 2011. – 158 с.

2. Ларионов С.А., Майоров Г.И. Вычислительные комплексы, системы и сети. – СПб.: Энергоатомиздат, 1987. – 186 с.

3. Романовский И.В. Алгоритмы решения экстремальных задач. – М.: Наука, 1977. – 148 с.

4. Нейдорф Р.А., Жикулин А.А. Быстродействующая модификация алгоритма оптимизации решения // Труды конгресса по интеллектуальным системам и информационным технологиям «AIS-IT'14». Научное издание в 4-х томах. Т.1. – М.: Физматлит, 2014. – С. 15-22.

5. Кобак В.Г., Будиловский Д.М. Сравнительный анализ приближенных алгоритмов решения минимаксной задачи для однородных приборов // Вестн. Донск. гос. техн. ун-та, 2006. – № 4. – С. 327-334.

6. Нейдорф Р.А., Жикулин А.А. Исследование вариантов модификации приближенных алгоритмов решения однородных распределительных задач, повышающих их эффективность // Инновация, экология и ресурсосберегающие технологии (ИнЭРТ-2012): Труды X Междунар. науч.-техн. форума. – Ростов н/Д: ИЦ ДГТУ, 2012. – С. 370-375.

7. Нейдорф Р.А., Жикулин А.А. Селективно-перестановочный метод приближенного решения однородной распределительной задачи. Комбинационные перестановки // Известия ЮФУ. Технические науки.



Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТТИ ЮФУ, 2013. – № 7(144). – С. 167-172.

8. Dorigo M., Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA, 2004. – 226 p.

9. Лебедев О.Б. Модели адаптивного поведения муравьиной колонии в задачах проектирования. Монография. -Таганрог. Изд-во ЮФУ, 2013. – 199 с.

10. Raidl G.R. A Unified View on Hybrid Metaheuristics. In: Lecture Notes in Computer Science. Springer- Verlag, 2006. – pp. 1-12.

11. Лебедев Б.К., Лебедев О.Б. Моделирование адаптивного поведения муравьиной колонии при поиске решений, интерпретируемых деревьями // Известия ЮФУ. Изд-во ТТИ ЮФУ, 2012 -№7. – С. 27-35.

12. Лебедев В.Б., Лебедев О.Б. Роевой интеллект на основе интеграции моделей адаптивного поведения муравьиной и пчелиной колоний // Известия ЮФУ. Изд-во ТТИ ЮФУ, 2013, №7. – С. 41-47.

13. Курейчик В.М., Лебедев Б.К., Лебедев О.Б. Разбиение на основе моделирования адаптивного поведения биологических систем. Нейрокомпьютеры: разработка, применение. 2010. -№2. – С. 28-34.

14. Карпенко А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновленные природой. – Москва: Издательство МГТУ им. Н.Э. Баумана, 2014. – 446 с.

15. Засядко Г.Е. Представление графовых моделей данных в виде n-арных деревьев // Инженерный вестник Дона, 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4166.

16. Пшихопов В.Х., Шевченко В.А., Медведев М.Ю., Гуренко Б.В. Управление распределенными системами подводной робототехники с использованием адаптивной эталонной модели // Инженерный вестник Дона, 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4130.

References

1. Lazarev A.A., Gafarov E.R. Teoriya raspisaniy. Zadachi i algoritmy. [Theory of schedules. The tasks and algorithms]. Moscow: Moscow State University M.V. Lomonosov. 2011. 158 p.
2. Larionov S.A., Mayorov G.I. Vychislitel'nyye kompleksy, sistemy i seti. [Computing systems, systems and networks]. St. Petersburg: Energoatomizdat. 1987. 186 p.
3. Romanovskiy I.V. Algoritmy resheniya ekstremal'nykh zadach. [Algorithms for solving extremal problems]. Moscow: Nauka. 1977. 148 p.
4. Neydorf R.A., Zhikulin A.A. Trudy kongressa po intellektual'nym sistemam i informatsionnym tekhnologiyam «AIS-IT'14». Nauchnoye izdaniye v 4-kh tomakh. T.1. [Proceedings of the Congress on Intelligent Systems and Information Technologies «AIS-IT'14». Scientific publication in 4 volumes. vol.1.] Moscow: Fizmatlit. 2014. pp. 15-22.
5. Kobak V.G., Budilovski, D.M. Vestn. Donsk. State. Tech. Un-ta. 2006. pp. 327-334.
6. Neydorf R.A., Zhikulin A.A. Trudy X Mezhdunar. nauch.- tekhn. Forum: Innovatsiya, ekologiya i resursosberegayushchiye tekhnologii (InERT-2012) [Innovation, Ecology and Resource-Saving Technologies (InERT-2012): Proceedings of the Xth International. Scientific-techn. Forum]. Rostov on Don: DSTU. 2012. pp. 370-375.
7. Neydorf R.A., Zhikulin A.A. Izvestiya SFU. Tekhnicheskiye nauki. Tematicheskiy vypusk «Intellektual'nyye SAPR» [Izvestiya SFU. Technical science. Thematic issue "Intellectual CAD"]. Taganrog: SFU, № 7 (144). 2013. pp. 167-172.
8. Dorigo M., Stützle T. Ant Colony Optimization. MIT Press, Cambridge, MA. 2004. 226 p.



9. Lebedev O.B. Modeli adaptivnogo povedeniya murav'inoy kolonii v zadachakh proyektirovaniya [Models of adaptive behavior of ant's colony in design problems]. Taganrog SFU. 2013. 199 p.

10. Raidl G.R. A Unified View on Hybrid Metaheuristics. In: Lecture Notes in Computer Science. Springer-Verlag. 2006. pp.1-12.

11. Lebedev B.K., Lebedev O.B. Izvestiya SFU. Tekhnicheskiye nauki. Tematicheskiy vypusk «Intellectual'nyye SAPR» [Izvestiya SFU. Technical science. Thematic issue "Intellectual CAD"]. Taganrog: SFU, № 7. 2012. pp. 27-35.

12. Lebedev V.B., Lebedev O.B. Izvestiya SFU. Tekhnicheskiye nauki. Tematicheskiy vypusk «Intellectual'nyye SAPR» [Izvestiya SFU. Technical science. Thematic issue "Intellectual CAD"]. Taganrog: SFU, № 7. 2013. pp. 41-47.

13. Kureychik V.M., Lebedev B.K., Lebedev O.B. Neyrokomp'yutery: razrabotka, primeneniye [Neurocomputers: development, application]. Moscow: №2, 2010. pp. 28-34.

14. Karpenko A.P. Sovremennyye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennyye prirodoy [Modern algorithms of search optimization. Algorithms inspired by nature]. Moscow: MSTU N.E. Bauman. 2014. 446 p.

15. Zasyadko G.E. Inzhenernyj vestnik Dona (Rus), 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4166.

16. Pshikhopov V.Kh., Shevchenko V.A., Medvedev M.Yu., Gurenko B.V. Inzhenernyj vestnik Dona (Rus), 2017, №2. URL: ivdon.ru/ru/magazine/archive/N2y2017/4130.