

компонентного анализа (ПКА). Это уменьшает не только количество правил (например, КНС), но также и количество функций принадлежности согласно предварительно заданным требованиям динамической точности. В изучении параметров ПНКНС применяется упорядоченное производное исчисление для получения периодических правил обучения, благодаря которым достигается возвратная структура ПНКНС, унаследованная от КНС. Выведенные правила могут рационально запоминать шаблоны КНС и другие параметры ПНКНС. Предлагаемые ПНКНС обеспечивают решение текущей задачи при создании шаблонов и/или нечетких правил существующей встроенной (нечеткой) системы КНС. Это может найти широкое применение при построении прогнозных моделей поведения сложных динамических систем, содержащих многочисленные изменяющиеся во времени параметры при взаимодействии с различными характеристиками окружающей среды. Экспериментальные результаты демонстрируют, что предлагаемая схема является эффективной и перспективной [8].

**Заключение.** В результате можно сделать вывод, что для сокращения времени разработки новой технической системы, повышения достоверности и оперативности контроля механических свойств компонентов ее конструкции целесообразно решать задачи синтеза программ испытаний прогнозных моделей в нейросетевом базисе.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прохорович В.Е. Прогнозирование состояния сложных технических комплексов. – СПб.: Наука, 1999.
2. Растринин Л.А., Пономорев Ю.П. Экстраполяционные методы проектирования и управления. – М.: Машиностроение, 1986.
3. Оссовский С. Нейронные сети для обработки информации / Пер. с польского И.Д. Рудинского. – М.: Финансы и статистика, 2002.
4. Назаров А.В., Лоскутов А.И. Нейросетевые алгоритмы прогнозирования и оптимизации систем. – СПб.: Наука и Техника, 2003.
5. I. Szatmari, D. Balya, G. Timar, C. Rekeczky, and T. Roska. Multi-channel spatio-temporal topographic processing for visual search and navigation. in Proc. SPIE Microtechnologies for the New Millenium, Gran Canaria, Spain, May 2003, Paper 5119-38.
6. F. Colodro and A. Torralba. Cellular neuro-fuzzy networks (CNFNs), a new class of cellular networks. in Proc. 5th IEEE Int. Conf. Fuzzy Systems, vol. 1, Sept. 8-11, 1996, pp. 517-521.
7. Cs. Rekeczky, T. Roska, and A. Ushida. CNN-based difference-controlled adaptive nonlinear image filters. Int. J. Circuit Theory Applicat., vol. 26, 1998, pp. 375-423.
8. C. T. Lin, C. L. Chang, and W. C. Cheng. A recurrent fuzzy neural network system with automatic structure and template learning. IEEE Transactions on circuits and systems – I: Regular papers, vol. 51, no. 5, May 2004, pp. 1024-1035.

**В.А. Литвиненко, С.А. Ховансков**

#### ОРГАНИЗАЦИЯ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ МУЛЬТИАГЕНТНОГО ПОДХОДА

При решении задач требующих обработки большого объема данных возникает проблема получения решения за приемлемое время. Одним из путей выполнения ограничения на временной ресурс является разделение всего объема вычислений между несколькими центрами обработки данных [1]. Обычно в качестве центров обработки используются работающие совместно персональные компьютеры.

При организации совместной работы компьютеров неизбежна проблема разделения задачи на вычислительные блоки между центрами обработки. Традиционно она решается созданием управляющей программы. Управляющая программа закрепляет между компьютерами вычислительные блоки задачи, организывает их совместную работу, следит за работоспособностью компьютеров, обеспечивает сбор обработанных данных [2].

Такой подход накладывает жесткие ограничения на круг выполняемых задач из-за невозможности быстрой перенастройки компьютеров в случае необходимости выполнения ряда задач с разными вычислительными алгоритмами. Также недостатками различных реализаций такого подхода являются неизбежные временные потери, большие затраты на разработку программы и низкая живучесть системы из-за наличия центра управления, низкая эффективность решения вычислительных блоков задач с высокой степенью связности по данным.

В настоящее время активно развивается направление, связанное с организацией процесса принятия решения, основанное на распределении объемов решаемой задачи с помощью технологии мобильных агентов, называемой мультиагентной системой (МС). Достоинствами технологии мобильных агентов является высокая надежность и возможность поддержки независимых вычислительных процессов [3].

В работе [4] предложено организовать распределенные вычисления части вычислительных блоков задачи с слабыми связями по данным, но наличие центра управления и отдельных центров хранения информации о компьютерах дают частичный выигрыш от применения МС.

В настоящей статье предлагается алгоритм работы МС для организации распределенных вычислений на нескольких персональных компьютерах (ПК) методом принятия коллективного решения.

При организации распределенных вычислений система решает проблему конфигурирования вычислительного процесса в компьютерной среде. Предложенный алгоритм предусматривает не только организацию совместной работы имеющихся центров обработки информации, но и предпринимает попытки увеличения количества центров обработки данных, оптимизирует получаемую конфигурацию в ходе выполнения компьютерами задачи с целью сокращения времени решения и повышения степени живучести всей системы.

Предлагаемый алгоритм работы МС позволяет использовать в качестве центров обработки не специализированные машины с необходимым аппаратным обеспечением, а обычные ПК, что уменьшает затраты и увеличивает живучесть вычислительного процесса. В организуемой системе нет требований к быстродействию компьютеров, а также не существует ограничения на количество ПК. Необходимым условием является наличие связей между ПК для передачи данных.

Это может быть и обычная компьютерная сеть, построенная по топологиям, имеющим иерархическую структуру, используя различные сетевые технологии (Ethernet). При такой конфигурации и сетевой технологии каждый ПК может обращаться к источнику входных данных по каналу связи независимо от других ПК, передавать данные и служебную информацию от компьютера к компьютеру.

Сеть используется при выполнении задач для получения обрабатываемой информации (входные данные) из одного источника во все ПК (например, сеть Интернет, решение задач САПР, задачи управления и др.). Все результаты обработок передаются по сети и собираются на одном из ПК.

Модуль МС (агент) – это программа, устанавливаемая на каждый компьютер. Включает в себя вычислительные модули, необходимые для выполнения самой

задачи обработки данных, и управляющую программу агента, реализующую предлагаемый алгоритм. Агент обладает правом использовать все вычислительные ресурсы своего ПК находящегося в сети. При использовании ПК аппаратные ресурсы и маршруты передачи данных между агентами образуют иерархическую вычислительную структуру. Входные данные поступают агентам всех уровней структуры, а выходные данные собираются у агента самого верхнего уровня. Агент самого верхнего уровня располагается на ПК пользователя.

Первоначально модуль агента может располагаться на одном или нескольких ПК. Все агенты являются копиями одного агента и работают по одному и тому же алгоритму. Агенты равноправны и управляют работой своего компьютера независимо друг от друга. Под управлением своего агента ПК занимается сбором и обработкой информации. По окончании выполнения заданного объема вычислений выполняется обмен между агентами ПК имеющими связи согласно иерархической структуре.

В ходе работы каждый агент старается увеличить количество агентов в сети и начинает поиск свободных от других агентов ПК. При обнаружении таких ПК выполняется их «вербовка». При этом происходит передача от агента-родителя своей копии в виде управляющего модуля агента-клона, который будет работать на «завербованном» ПК. В случае передачи от агента-клона обработанных данных номер ПК запоминается как завербованный. В случае отсутствия передачи ПК запоминается как занятый. Появившийся агент-клон в иерархической структуре будет принадлежать следующему нижнему уровню МС по отношению к агенту-родителю.

Каждый агент-клон обменивается выходными данными только с тем агентом, который для него является родителем. Разрастаясь таким образом, иерархическая структура вычислительной системы может иметь неограниченное количество уровней.

Агент-клон является полной копией породившего его агента-родителя и будет выполнять те же функции, что и агент-родитель: управление обработкой данных; обмен ею с другими агентами; передача служебной информации агентам-родителям и своим агентам-клонам; поиск и вербовка новых свободных ПК.

Количество агентов-клонов у порождаемого их агента-родителя может быть бесконечно. Все множество агентов делится на совокупности агентов-родителей и их агентов-клонов:

$$l_{\beta}^{i+1}, l_{\beta+1}^{i+1}, \dots, l_{\beta+(k-1)}^{i+1} \in l_{\alpha}^i \quad (1)$$

где

$l_{\alpha}^i$  – ПК на  $i$ -ом уровне, на котором расположен агент-родитель.

Каждый агент-родитель, имея агентов-клонов, передает им вместе с служебной информацией сведения, содержащие данные о наиболее коротком маршруте передачи выходных данных и служебной информации от агента-клона агенту самого высокого уровня.

Для повышения живучести системы информация о маршруте используется в случае выхода из строя или отключения ПК, входящих в маршрут передачи данных. Если из указанного маршрута выбыл или отключился ПК, то связанный с ним агент-клон более нижнего уровня передает информацию агенту ближайшего подключенного ПК высокого уровня, присутствующего в указанном маршруте передачи результата.

При многоуровневой иерархии агент-клон может быть одновременно и агентом-родителем.

При работе агентов весь процесс работы МС делится на одинаковые по действиям каждого агента шаги. Под шагом понимается повторяющийся набор действий: прием данных, их обработка, прием и обработка служебной информации, ее передача и передача своего и принятого результатов.

Шаг начинается с того, что после обработки данных каждый агент-клон передает агенту являющимся его родителем помимо выходных данных еще и служебную информацию о ПК, на которых эти вычисления выполнялись:

- ◆ скорость выполнения данным ПК заданного объема вычислений;
- ◆ время получения выходных данных и передачи;
- ◆ время приема обработанных данных.

Это «прямая» связь в МС «снизу-вверх». Позволяет осуществить сбор результатов вычислений при любом количестве центров обработки данных. Существует и «обратная» связь «сверху-вниз». Она выполняется после передачи информации по «прямой» связи. Все агенты-клоны после своей передачи информации по прямой связи ждут передачи служебной информации от своих агентов-родителей.

После получения результатов от агентов нижнего уровня агентами-родителями анализируются служебные данные, полученные при «прямой» передаче. При «обратной» передаче от агента-родителя высокого уровня всем связанным с ним агентам-клонам нижнего уровня будет передана служебная информация содержащая, если это нужно, управляющие данные о назначении агенту-клону объема вычислений выполняемых при обработке входных данных и путь передачи выходных данных на самый верхний уровень.

Каждый агент обменивается на каждом шаге информацией с агентом верхнего уровня иерархической структуры, являющимся его родителем, и своими агентами-клонами нижнего уровня, которые поддерживают с ним связь. По этим связям агент занимается не только приемом и передачей обработанных данных на верхний уровень, но и формированием и передачей служебной информации на верхний и нижний уровни структуры. Для агента время шага складывается из времени приема входных данных, выполнения своего объема вычислений, времени обработки и передачи различной информации агентам соседних уровней.

Основной задачей для агента остается выполнение своего объема вычислений решаемой задачи, какой бы уровень он ни занимал.

Агент-родитель при появлении своего нового агента-клона пытается перераспределить весь объем обрабатываемых данных, который назначен ему агентом-родителем на части примерно в равной пропорции между своими агентами-клонами, с учетом своих собственных вычислительных ресурсов. При последующих шагах происходит корректировка объема обрабатываемых данных им и его клонами с учетом вычислительных ресурсов каждой ПК. Сам агент-клон являясь агентом-родителем также изменяет объем заданий своим агентам-клонам в соответствии с установленной от своего агента-родителя долей от общего объема обрабатываемых данных.

Это позволяет уменьшить разницу во времени получения результатов решения на всех ПК МС одного уровня и повышает надежность работы системы. Выравнивание может выполняться и по другим критериям в зависимости от объема вычислений и пропускной способности каналов связи.

Обозначим  $P$  – весь объем обрабатываемой информации на одном шаге. В самом начале работы МС, структура имеет один уровень и агент работает только на одном ПК  $nI = I$ . Соответственно весь объем вычислений  $P$  выполняется агентом на этом ПК. При расширении структуры каждый ПК присоединяется к

соответствующему уровню  $l_{\alpha}^i \in nI$ . Агенты второго уровня образуют множество  $n2 = (l_1^2, l_2^2, \dots, l_m^2)$ .

При размножении агента на другие ПК весь объем вычислений (2) начинает делиться между всеми агентами МС:

$$P_{\alpha}^i = P_{\beta}^{i+1} + P_{\beta+1}^{i+1} + \dots + P_{\beta+(k-1)}^{i+1} + P_{\alpha}^i, \quad (2)$$

где  $k$  – количество агентов-клонов агента-родителя ПК  $l_{\alpha}^i$

Время решения задачи на одном шаге (3) определяется наибольшим временем выполнения своего объема вычислений этого шага одним из агентов входящих в совокупность агент-родитель и его агенты-клоны:

$$t_{\alpha}^i = \max(t_{\beta}^{i+1}, t_{\beta+1}^{i+1}, \dots, t_{\beta+(k-1)}^{i+1}). \quad (3)$$

Для второго уровня:

$$t^1 = \max(t_1^2, t_2^2, \dots, t_m^2). \quad (4)$$

Среднее время вычислений агентами-клонами:

$$t_{cp \alpha}^i = t_{\beta}^{i+1} + t_{\beta+1}^{i+1} + \dots + t_{\beta+(k-1)}^{i+1} = \frac{\sum_{j=0}^{j=k-1} t_{m+j}^{i+1}}{k}. \quad (5)$$

Агент-родитель перераспределяет между своими агентами клонами объемы производимых вычислений для уменьшения максимального времени выполнения агентами-клонами своих задач:

$$(P_m^{i+1})' = P_m^{i+1} \frac{t_{cp \alpha}^i}{t_m^{i+1}}. \quad (6)$$

#### Алгоритм агента.

1. Получение входных данных.
2. Запуск выполнения обработки входных данных.
3. Проверить – адрес присутствующего в сети компьютера есть в списке проверенных компьютеров? Если адрес есть, то перейти к п.5, иначе перейти к п.4.
4. Передача обнаруженному компьютеру копии агента.
5. Проверить – выполнена ли обработка входных данных? Если обработка выполнена, то перейти к п.6, иначе перейти к п.3.
6. Проверить – получены ли выходные данные? Если данные получены, то перейти к п.7, иначе перейти к п.3.
7. Запись полученных данных.
8. Проверить – получены ли данные от агентов уровня  $i-1$ ? Если данные получены, то перейти к п.9, иначе перейти к п.10.
9. Занесение компьютера-отправителя в список агентов уровня  $i-1$ .
10. Проверить – все ли агенты уровня  $i-1$  передали данные? Если все, то перейти к п.11, иначе перейти к п.3.
11. Передача всех выходных данных.
12. Проверить – получена ли служебная информация от агента уровня  $i+1$ ? Если получена, то перейти к п.14, иначе перейти к п.13.
13. Передача всех выходных данных следующему агенту согласно маршрута передачи данных.

14. Расчет  $t_{cp}^i$  и  $P_m^{i+1}$ .

15. Передача служебной информации агентам уровня  $i - 1$ .

16. Переход к п.1.

Разработанный алгоритм организует и оптимизирует распределенные вычисления методом принятия коллективного решения с использованием МС. Выполняется на фоне решения основной задачи. Время, затраченное на переконфигурацию структуры МС, из-за изменений в системе или поступления нового задания сводится к минимуму и оказывает относительно небольшое влияние на общее время решения задачи.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лорин Г. Распределенные вычислительные системы. – М.: Радио и связь, 1984. – 296 с.
2. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.
3. Ховансков С.А., Мельник Э.В. Блушвили И.В. Метод организации распределенных вычислений в управляющих системах // Мехатроника, автоматизация, управление. «Новые технологии», №4, 2003.
4. Мельник Э.В., Иванов В.В., Погорелов К.В. Об одном подходе к организации поиска информации в Интернете // Известия ТРТУ, Тематический выпуск «Интеллектуальные и многопроцессорные системы». – Таганрог: Изд-во ТРТУ, №10, 2005.

**М.В. Щербинина**

#### ПРИМЕНЕНИЕ НЕЙРОСЕТЕВОЙ СТРУКТУРЫ МОДУЛЬНОГО ТИПА ДЛЯ РЕШЕНИЯ НАВИГАЦИОННОЙ ЗАДАЧИ\*

Ставится задача определения координат летательного аппарата по измеренным расстояниям до опорных радиомаяков. Если рассматривается более трех радиомаяков, то аналитически задача не решается. Предложенные способы решения, основанные на методе наименьших квадратов, требуют больших вычислительных ресурсов. Поэтому предлагается использовать нейронные сети для решения поставленной задачи. Нейросетевые средства отличаются компактностью вычислений, при этом аппроксимируют желаемый результат с достаточно высокой точностью. В работе предлагается использовать модульную нейросетевую структуру для решения навигационной задачи.

**Постановка задачи.** Навигационная задача формулируется следующим образом (рис.1). В трёхмерном декартовом пространстве заданы координаты точек, в которых находятся опорные радиомаяки. Необходимо определить положение летательного аппарата (ЛА) по измеренным расстояниям  $D_i$  от ЛА до опорных радиомаяков ( $i = \overline{1, n}$ ). Предполагается, что схема расположения маяков фиксирована, а ЛА находится внутри заданного куба, называемого областью решения. Необходимо с точностью не более чем 10 м оценить координаты летательного аппарата  $x$ ,  $y$  и  $z$  по измерениям дальностей  $D_i$  до радиомаяков. Предполагается, что координаты радиомаяков заданы с погрешностями и измерения дальностей также производятся на фоне помех.

Поставленная прикладная задача относится к классу задач аппроксимации функций многих переменных. В роли функций выступают оцениваемые координаты

\* Работа выполнена при финансовой поддержке РФФИ (проект № 05-08-01421).