

На правах рукописи

Погребной Александр Владимирович

Математические и программные средства построения архитектуры и топологии сети вычислительной системы для управления территориально распределенными объектами

05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

А в т о р е ф е р а т

**диссертации на соискание ученой степени
кандидата технических наук**

Томск – 2008

Работа выполнена в Томском политехническом университете

Научный руководитель: доктор технических наук, профессор
Силич Виктор Алексеевич

Официальные оппоненты: доктор технических наук, профессор
Марков Николай Григорьевич

кандидат технических наук, доцент
Хабибулина Надежда Юрьевна

Ведущая организация: Новосибирский государственный тех-
нический университет

Защита состоится «24» декабря 2008 г. в 14-30 часов на заседании Сове-
та по защите докторских и кандидатских диссертаций Д 212.269.06 при Том-
ском политехническом университете по адресу: г. Томск, ул. Советская 84, ин-
ститут «Кибернетический центр» ТПУ.

С диссертацией можно ознакомиться в Научно-технической библиотеке Том-
ского политехнического университета по адресу: г. Томск, ул. Белинского 55.

Автореферат разослан «__» _____ 2008 г.

Ученый секретарь диссертационного
совета кандидат технических
наук, доцент

_____ М.А. Сонькин

Общая характеристика работы

Актуальность работы. С развитием микропроцессорной и телекоммуникационной техники появилась возможность размещать средства обработки информации вблизи мест зарождения исходных данных и их использования. Это обстоятельство сделало возможным создание эффективных систем управления объектами с территориально распределенным оборудованием. Территории, на которых распределено оборудование объектов, могут измеряться десятками и сотнями метров и несколькими километрами. Управление прокатным станом, роботами, движением на магистралях, контроль за состоянием окружающей среды, управление химическими, энергетическими установками и многими другими объектами определяет предметную область систем управления такими объектами в реальном масштабе времени. Системы реального времени (СРВ) становятся неотъемлемой частью современных высокотехнологичных промышленных систем.

Теоретические основы методов проектирования СРВ к настоящему времени разработаны слабо. Основные научные интересы разработчиков СРВ были сосредоточены на создании операционных систем и инструментальных средств программирования. К настоящему времени разработаны десятки коммерческих и специализированных операционных систем реального времени. Интенсивно развиваются объектно-ориентированные методологии разработки программного обеспечения.

Поскольку, в последнее время, все большее число СРВ разрабатывается как распределенные и сетевые, в составе инструментальных средств для проектирования таких систем возросла роль задач, связанных с определением структуры сети и ее топологической привязки к оборудованию распределенного объекта. Данные задачи до настоящего времени не привлекали должного внимания исследователей и продолжают оставаться малоизученными. Действительно, эти задачи непосредственно не затрагивают стадию создания технологий разработки программного обеспечения СРВ, но применение технологии при проектировании конкретной СРВ может осуществляться только при наличии результатов решения данных задач. При этом должны быть получены ответы на следующие вопросы:

- число микропроцессорных станций, которое понадобится для своевременного выполнения прикладных функций проектируемой системы;
- как эти станции размещены на территории расположения объекта и какие терминальные точки (датчики и исполнительные механизмы) подключены к ним;
- как связаны между собой станции в единую локальную сеть.

В известных подходах к проектированию СРВ вначале решается задача определения числа станций, после этого вручную принимается вариант топологии сети и далее решается задача распределения программной нагрузки по станциям чаще всего по критерию минимальной загрузки сети. Предлагаемые при этом оптимизационные постановки задач характеризуются большой раз-

мерностью, нелинейностью и оказываются непригодными для решения практических задач.

Второй недостаток заключается в том, что при решении задачи получения плана использования ресурсов, топология сети принимается заданной. Существенным недостатком является также и то, что не учитываются условия функционирования объекта управления, такие как условия поступления входных данных и, соответственно, запуска процессов, условия обновления состояний выходных данных, правила селекции состояний выходных данных при передаче их между процессами. Перечисленные условия непосредственно влияют на объем данных, передаваемых в сети.

Изложенное выше показывает, что в области проектирования СРВ разработано большое разнообразие операционных систем реального времени с различными средствами разработки приложений реального времени – кроссовых, резидентных и комбинированных. Большое развитие получили методы моделирования и исследования по взаимодействию параллельных процессов. Что касается задач определения числа станций, их размещения на территории объекта и построения сети, то анализ известных подходов выявил существенные недостатки, которые не позволяют сформировать целостную совокупность проектных процедур для решения задач построения архитектуры и топологии сети вычислительной системы проектируемой СРВ.

Предлагаемый в диссертации подход к построению качественной исходной модели вычислительной системы наряду с топологией размещения оборудования максимально учитывает характеристики модели программной нагрузки и условий функционирования объекта управления. Определение объемов данных, передаваемых в сети, и выбор на этой основе топологии сети выделены в отдельные задачи, которые особенно важны при проектировании жестких СРВ. Полученные при этом постановки задач имеют размерности, приемлемые для практического применения.

Цель работы и задачи исследования. Целью диссертации является разработка методов и программных средств, которые на основе заданной топологии расположения объекта управления и условий его функционирования, позволяют аналитическим путем определить архитектуру и топологию сети многопроцессорной вычислительной системы для проектируемой СРВ по управлению данным объектом.

Для достижения поставленной цели были поставлены и решены следующие задачи:

- определение числа станций вычислительной системы исходя из заданного количества терминальных точек по сбору и использованию данных и условий функционирования объекта управления;
- определение мест размещения станций вычислительной системы на территории расположения объекта управления;
- распределение (подключение) терминальных точек по станциям по критерию минимума суммы расстояний от точек до станций;

- разработка методики оценки потребности программной нагрузки СРВ в сетевых ресурсах и на этой основе определение структуры локальной сети вычислительной системы;
- разработка программных средств для решения перечисленных задач, проведение экспериментальных исследований соответствующих методов и их применение для решения практических задач.

Методы исследований. При проведении исследований и разработке математического и программного обеспечения для решения перечисленных выше задач использованы методы линейного математического программирования, теории графов, взаимодействия параллельных процессов, расписаний, распараллеливания моделей алгоритмов, локальных вычислительных сетей, моделирования, алгоритмизации и объектно-ориентированного программирования.

Научную новизну имеют следующие результаты:

1. Постановка задачи определения числа станций вычислительной системы в виде целочисленной задачи линейного математического программирования.
2. Метод решения задачи размещения станций на территории расположения объекта управления.
3. Постановка задачи распределения терминальных точек по станциям как задачи линейного математического программирования транспортного типа.
4. Алгоритм получения на множестве терминальных точек локальных компактных разбиений, введение ряда оценок компактности и исследование их свойств.
5. Метод анализа модели программной нагрузки, представленной в форме графа потока данных, и построения на этой основе структуры локальной сети вычислительной системы.

Практическая ценность результатов работы

Разработанное в диссертации математическое и программное обеспечение в составе программных средств «Полюс», «Топология», «Сеть», «Редактор архитектур» позволяет разработчикам распределенных СРВ спроектировать исходный вариант архитектуры и топологии сети многопроцессорной вычислительной системы.

Существенным достоинством разработанных алгоритмов и программ является то, что они наглядны, эффективны и способны решать задачи, размерность которых и достигаемая при этом точность приемлемы для практического применения. Сюда относятся следующие задачи: определения числа станций (задача покрытия); формирования исходных множеств терминальных точек для подключения к станциям (алгоритмы $\alpha_1, \alpha_2, \alpha_3$); размещения станций (алгоритм по схеме метода ветвей и границ); распределения терминальных точек по станциям (транспортные задачи T и T^*); получения локальных компактных разбиений (процедура, включающая задачу T); формализованного перехода от модели программной нагрузки СРВ к структуре локальной сети вычислительной системы; построения контуров обхода станций при их обслуживании.

Результаты исследований позволили построение архитектуры и топологии сети вычислительной системы поставить в зависимость от значений параметров модели программной нагрузки и условий функционирования объекта управления. Эту зависимость удалось установить аналитическими методами на начальной стадии проектирования до привлечения трудоемких систем моделирования.

Полученные результаты разработаны в составе системы автоматизированного синтеза СРВ, но могут быть использованы автономно в проектных работах по внедрению информационных технологий в территориально распределенных технических системах в различных предметных областях. В работе это показано на примере наземной метеорологической наблюдательной сети Росгидромета.

Программы «Полюс» и «Топология» могут быть полезны при решении широкого круга задач, связанных с разбиением множеств объектов на подмножества и с разрезанием графов на минимально связанные подграфы.

Опыт использования в учебном процессе программных средств «Полюс», «Топология», «Сеть», «Редактор архитектур» при выполнении курсовой работы по проектированию СРВ показал, что через решение взаимосвязанной совокупности конкретных задач у студентов формируется ясное представление о взаимосвязях параметров и характеристик в такой сложной системе, какой является распределенная СРВ.

Основные положения выносимые на защиту:

1. Исследования показали, что из модели программной нагрузки, представленной в форме графа потока данных, и условий функционирования объекта управления можно извлечь сведения, достаточные для принятия решений по архитектуре и топологии сети вычислительной системы для проектируемой распределенной СРВ.

2. В условиях эволюционного проектирования и использования интерактивного сценария принятия решений, предпочтительным является разбиение процесса построения архитектуры и топологии сети вычислительной системы на совокупность простых и автономно выполняемых проектных процедур. Применение существующих стратегий совместного решения сложных взаимосвязанных задач, требуют больших объемов вычислений, а получаемые при этом решения после очередной вынужденной трансформации модели вычислительной системы оказываются бесполезными.

3. Разработанная в диссертации совокупность проектных процедур создает основу математического и программного обеспечения инструментальных средств, которые поддерживают интерактивный режим эволюционного проектирования и путем аналитических вычислений позволяют получить качественный исходный вариант архитектуры и топологии сети вычислительной системы.

4. Метод получения локальных компактных разбиений на основе итерационной процедуры применения задачи математического программирования транспортного типа и анализа оценки разбиения, является эффективным инст-

рументом, который может быть использован во многих приложениях при поиске разбиений, компактных относительно принятых оценок.

5. При заданном числе станций величина минимального объема передаваемых в сети данных является объективной характеристикой информационной модели программной нагрузки и соответствует оптимальному разбиению информационного графа на части. Выбирая вариант топологии сети и план подключения станций к магистралям, можно за счет максимизации объемов параллельно передаваемых данных, существенно сократить суммарное время на передачу данных в сети.

Апробация работы. Основные результаты работы докладывались и обсуждались на следующих конференциях: Восьмой Российско – Корейский международный симпозиум по науке и технологии (The 8 Russian–Korean International Symposium on Science and Technology) «KORUS 2004» (г. Томск, ТПУ, 2004г). Конференция «Молодежь и современные информационные технологии» (г. Томск, ТПУ, 2004г). Девятый Российско-Корейский международный симпозиум по науке и технологии (The 9 Russian–Korean International Symposium on Science and Technology) «KORUS 2005» (г. Новосибирск, НГТУ, 2005г). Международная конференция «Современная техника и технологии» (г. Томск, ТПУ, 2006г). Международная конференция «Современная техника и технологии» (г. Томск, ТПУ, 2007г).

По результатам исследований опубликовано 5 докладов на конференциях и 5 статей, три из них в изданиях рекомендованных ВАК.

Структура и объем работы. Диссертация состоит из введения, четырех глав, заключения, списка литературы из 96 наименований. Каждая глава сопровождается выводами. Общий объем диссертации составляет 171 страницу машинописного текста, включает 67 рисунков и 6 таблиц.

Основное содержание работы

Во введении приводится анализ состояния исследований по формализации методов построения исходной модели архитектуры и топологии сети вычислительной системы при эволюционном проектировании распределенной СРВ. Отмечаются недостатки существующих подходов – большая размерность и нелинейность задач, не учитываются многие условия функционирования объекта управления, не рассматривается задача выбора топологии сети. Исходя из этого сформулированы цели и задачи исследования, показана их актуальность. Выделены результаты, имеющие научную новизну, показана практическая ценность работы и приведены основные положения, выносимые на защиту.

Первая глава посвящена определению числа станций вычислительной системы. Здесь приводится описание предметной области, дается определение территории расположения объекта управления, названной топологическим полем (ТП), терминальных точек, станций и их ресурсов, модели программной нагрузки, вычислительных процессов реального времени, цикла моделирования.

Основную часть программной нагрузки на вычислительную систему составляют алгоритмы выполнения прикладных функций. В качестве исходного описания модели программной нагрузки принят способ представления алго-

ритмов в форме графа потока данных (ГПД). В ГПД два вида вершин – алгоритмы или их фрагменты (модули) и данные. Дуги в графе связывают модули и данные и указывают на отношение модулей к потреблению и формированию тех или иных данных. Форма ГПД используется в работе также для представления модели алгоритма функционирования объекта управления.

Выделены два фактора, влияющие на необходимое число станций – множество терминальных точек расположенных на ТП и характеристики модели программной нагрузки. Первый фактор связан с выбором минимального числа станций способных подключить все терминальные точки ТП.

Совокупность терминальных точек ТП обозначим множеством E ,

$E = \bigcup E_r, \bigcap E_r = \emptyset, |E_r| = b_r, r = 1, 2, \dots, R$, где E_r – подмножество точек r -го типа в множестве E ; b_r – число точек r -го типа в множестве E ; R – число типов точек на ТП.

Число видов станций, которые могут быть использованы для построения вычислительной системы, обозначим величиной S . Для станций s -го вида введем вектор подключения точек $A_s = \{a_{rs}\}$, $r = 1, 2, \dots, R, s = 1, 2, \dots, S$, где a_{rs} – допустимое число мест подключения терминальных точек r -го типа в станции s -го вида.

Способность станций подключать терминальные точки согласно векторам A_s представим матрицей $A = \|a_{rs}\|_{R \times S}$, $r = 1, 2, \dots, R, s = 1, 2, \dots, S$.

Введем переменную $x_s, s = 1, 2, \dots, S$, которая определяет число станций s -го вида, используемых при подключении точек множества E . Тогда задачу определения минимального числа станций необходимых для подключения всех точек множества E можно записать в виде:

$$\sum_{s=1}^S x_s \Rightarrow \min; \quad (1)$$

$$\sum_{s=1}^S a_{rs} x_s \geq b_r, \quad r = 1, 2, \dots, R; \quad (2)$$

$$x_s - \text{положительное целое число для всех } s = 1, 2, \dots, S. \quad (3)$$

Задача (1)–(3) относится к классу задач целочисленного линейного программирования. Величины a_{rs} и b_r в ограничениях (2) – положительные числа, часто и целые, для многих приложений – булевские. Благодаря этой специфике, данная задача стала более известной как задача покрытия. В рассматриваемом приложении задачу можно представить как задачу покрытия терминальных точек множества E заданного вектором $B = \{b_r\}$, векторами подключения станций $A_s = \{a_{rs}\}$.

В результате решения задачи (1)–(3) получим вектор $X^* = \{x_s^*\}, s = 1, 2, \dots, S$, компоненты x_s^* которого определяют минимальное число станций s -го вида, необходимых для подключения терминальных точек множества E .

Для оценки влияния второго фактора необходимо проверить – способно ли выбранное число станций выполнить программную нагрузку по управлению объектом в заданное время. Предложено четыре этапа проверки способности вычислительной системы с заданным числом станций своевременно выполнить программную нагрузку.

Первый этап сопоставляет суммарные ресурсы выбранного числа станций, процессорное время, память и потребности модулей ГПД в этих ресурсах.

На втором этапе ГПД представляется совокупностью процессов и для каждого из них проверяется возможность завершения выполнения в установленном интервале времени.

Третий этап связан с проверкой возможности своевременного завершения выполнения прикладных функций СРВ в условиях представления ГПД в ярусно – параллельной форме.

Построение и исследование временной диаграммы в одном цикле моделирования функционирования программной нагрузки СРВ представлено в четвертом заключительном этапе проверки.

Результаты проверок формируют достаточно полное представление о том, как справляется вычислительная система с программной нагрузкой, и дают информацию для принятия решений по уточнению числа станций, планов использования их ресурсов, организации взаимодействия процессов и структуры локальной сети вычислительной системы.

Во второй главе предлагаются методы решения задачи размещения станций на ТП, распределения по ним терминальных точек, получения локальных компактных разбиений, приводятся результаты исследования некоторых свойств компактных разбиений.

Стратегия решения задачи размещения станций основана на формировании расширенной совокупности «хороших» множеств терминальных точек с последующим выбором предпочтительных множеств для размещения в них станций. Задача выбора предпочтительных множеств из расширенной совокупности сформулирована как задача линейного математического программирования с булевыми переменными. Для ее решения предложен алгоритм по схеме метода ветвей и границ.

Для формирования расширенной совокупности «хороших» множеств предложено три алгоритма $\alpha_1, \alpha_2, \alpha_3$. Каждый из них реализует некоторое эвристическое правило, настроенное на учет определенной специфики ТП конкретного объекта управления. Так, алгоритм α_1 ориентирован на ТП с множеством точек E одного типа и относительно равномерным расположением точек. Если множество E содержит несколько типов точек, то каждое из подмножеств E_r также должно быть распределено на ТП равномерно. Применение алгоритма α_2 более эффективно, если число типов точек R велико и условия равномерности распределения для множества E и подмножеств E_r не накладываются. Алгоритм α_3 занимает промежуточное положение между алгоритмами α_1 и α_2 , и отличается способом формирования множеств.

В основу алгоритма α_1 положено эвристическое правило, согласно которому вначале формируется совокупность точек $\pi_1 \subset E$, именуемых полюсами. Первый полюс, точка $e_{p_1} \in \pi_1$, выбирается из условия

$$\max_{e_{p_1} \in E} \left\{ \sum_{e_t \in E \setminus e_{p_1}} d(e_t, e_{p_1}) \right\}. \quad (4)$$

Здесь $d(e_t, e_{p_1})$ – расстояние от точки e_t до полюса e_{p_1} . Вторая и все последующие точки e_p множества π_1 выбираются по условию

$$d(e_t^*, e_p) = \max_{e_t \in E \setminus \pi_1} \min_{e_p \in \pi_1} d(e_t, e_p). \quad (5)$$

Точка e_t^* принимается в качестве очередного полюса e_p и включается в множество π_1 . Число полюсов соответствует числу станций P либо увеличивается разработчиком в K раз, где K , как правило, принимается равным числу R .

Для каждого p -го полюса в точке $e_p \in \pi_1$ и станций s -го вида, $s = 1, 2, \dots, S$, формируется множество E_{ps} , точки которого по типам соответствуют вектору подключения A_s станции s -го вида и удалены от полюса e_p на минимальные расстояния. Здесь S – число видов станций, для которых $x_s^* > 0$. В итоге для всех полюсов будет сформировано $|\pi_1|S$ множеств E_{ps} , $p = 1, 2, \dots, |\pi_1|$, $s = 1, 2, \dots, S$.

В алгоритме α_2 множества E_{ps} формируются непосредственно после определения очередного полюса e_p по выражению (4). Среди множеств E_{ps} , $s = 1, 2, \dots, S$ сформированных для данного полюса выбирается множество E_{ps}^* с минимальной оценкой

$$\left(\sum_{e_t \in E_{ps}} d(e_t, e_p) \right) / \sum_{r=1}^R a_{rs}. \quad (6)$$

Далее величина x_s^* уменьшается на 1 и из множества E исключается множество E_{ps}^* . Очередной полюс выделяется на множестве $E \setminus E_{ps}^*$. Последующее формирование множеств E_{ps} и выбор среди них множества с минимальной оценкой (6) осуществляется аналогично. Процесс заканчивается, когда для всех станций $\{x_s^*\}$ будут выделены полюса и соответствующие множества E_{ps}^* .

Заметим, что число полюсов $|\pi_2|$, выделяемых алгоритмом α_2 , равно числу станций P . При этом для каждого полюса формируется S множеств E_{ps} . Таким образом общее число множеств E_{ps} , формируемых алгоритмом α_2 , составит $|\pi_2|S$.

Алгоритм α_3 на первом этапе работает с множеством E , в котором типы точек не учитываются. Принимается также, что все станции имеют одинаковые

векторы подключения. Выбор первого полюса e_p осуществляется по выражению (4) и для него из ближайших точек формируется множество E_p , мощность которого принимается равной $|E| / \sum_{s=1}^S x_s^*$. Множество E_p исключается из множества E и процесс выбора второго полюса из точек $E \setminus e_p$ с формированием соответствующего множества E_p повторяется.

Множества E_p , сформированные по данному алгоритму, по составу и типам точек могут существенно отличаться от состава точек векторов подключения станций A_s . Поэтому на втором этапе алгоритма возникает необходимость в распределении станций по множествам E_p так, чтобы составы точек были максимально согласованы. С этой целью для каждого множества E_p по аналогии с вектором A_s формируется вектор состава точек A_p . Вводится оценка соответствия векторов A_s и A_p и путем решения задачи назначения, определяется наилучший вариант соответствия между множествами E_p и видами станций.

Число полюсов $|\pi_3|$, полученное по алгоритму α_3 , совпадает с числом станций P . Для каждого полюса $e_p \in \pi_3$, по аналогии с алгоритмами α_1 и α_2 , формируются множества E_{ps} и включаются в расширенную совокупность.

Таким образом с помощью алгоритмов $\alpha_1, \alpha_2, \alpha_3$ можно сформировать расширенную совокупность множеств E_{ps} . Число множеств в совокупности определяется величиной $\sigma = (|\pi_1| + |\pi_2| + |\pi_3|)S$. Например, если число станций $P = 6$, число видов станций S с $x_s^* > 0$ равно 4, а коэффициент кратности $K = R = 3$, то $\sigma = (RP + 2P)S = (3 * 6 + 2 * 6)4 = 120$.

Для каждого множества E_{ps} определяется «центр» e_c с минимальным суммарным удалением c_{ps} от точек $e_t \in E_{ps}$, $c_{ps} = \sum_{e_t \in E_{ps}} d(e_c, e_t)$. Величина

$c_{ps}^* = \max_{p,s} c_{ps} - c_{ps}$ принимается в качестве оценки множества E_{ps} .

Принадлежность точек $e_t \in E, t = 1, 2, \dots, m$ множествам E_{ps} представим матрицей $A = \|a_{tps}\|$, $t = 1, 2, \dots, m$, $p = 1, 2, \dots, n$, $s = 1, 2, \dots, S$. Здесь элемент $a_{tps} = 1$, если точка $e_t \in E_{ps}$, $a_{tps} = 0$, если $e_t \notin E_{ps}$, $m = |E|$, $n = |\pi| = |\pi_1| + |\pi_2| + |\pi_3|$.

Введем переменную $x_{ps} = 1$, если множество E_{ps} включено в решение, $x_{ps} = 0$ в противном случае. Тогда задачу выбора множеств E_{ps} по критерию максимальной суммы их оценок c_{ps}^* можно записать в виде:

$$\max L = \sum_{p=1}^n \sum_{s=1}^S c_{ps}^* x_{ps} ; \quad (7)$$

$$\sum_{p=1}^n \sum_{s=1}^S a_{tps} x_{ps} \leq 1, \quad t = 1, 2, \dots, m^* ; \quad (8)$$

$$\sum_{p=1}^n x_{ps} = x_s^*, \quad s = 1, 2, \dots, S. \quad (9)$$

Задача (7) – (9) относятся к классу задач линейного математического программирования с булевыми переменными и имеет специфику. Переменные x_{ps} ограничением (9) разбиваются на S групп. Для каждой группы заведомо известно число x_s^* переменных x_{ps} , которые будут выбраны для размещения станций.

Для решения данной задачи в диссертации предложен алгоритм по схеме метода ветвей и границ, в котором при оценке перспективности выбора вершин дерева решений, учитывается указанная специфика. В результате решения задачи будет выбрано P множеств E_{ps} , составы точек в которых соответствуют векторам подключения станций, определяемых вектором X^* . Размещение станций производится в центрах выбранных множеств E_{ps} .

Распределение точек по станциям, полученное в результате решения задачи (7) – (9) не может быть принято окончательным. В частности, условие (8) допускает подключение не всех точек множества E к станциям. Кроме того возможно перераспределение точек между станциями, улучшающее критерий (7). Исходя из этого после решения задачи (7) – (9) осуществляется оптимизация распределения точек по станциям. Замечено, что решение данной задачи можно выполнять отдельно для точек каждого подмножества E_r .

Число точек в множестве E_r для практических задач может измеряться сотнями. Поэтому в целях дальнейшего сокращения размерности задачи множество E_r предварительно разбивается на подмножества E_j , $j = 1, 2, \dots, m$; компактно расположенные на ТП, $\cup E_j = E$, $\cap E_j = \emptyset$, $E_j \neq \emptyset$. Компактность множества E_j будем оценивать величиной ξ_j , $\xi_j = L_j / |E_j|$. Среди возможных алгоритмов формирования множеств E_j с заданной граничной оценкой компактности ξ_j , воспользуемся способом разбиения, которое получается путем наложения на ТП регулярной сетки с фиксированным шагом, как это показано на рис. 1.

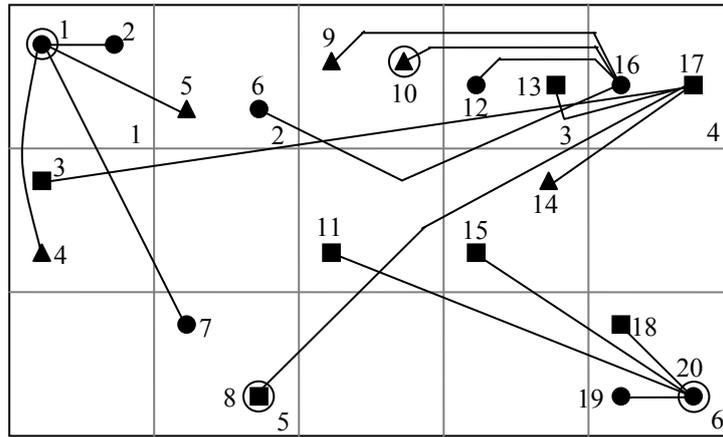


Рис. 1. Пример ТП с множествами E_{ps} и наложением сетки

Иллюстративный пример ТП, приведенный на рис. 1, содержит 20 точек трех типов, обозначенных символами (▲●■). Вектор X содержит три ненулевых компоненты $x_2^* = 2$, $x_1^* = x_3^* = 1$. Станции s_1 соответствует вектор подключения $A_1 = \{ \bullet\bullet\blacksquare\blacksquare\blacksquare\}$, станциям $s_2, s_3 - \{ \bullet\bullet\bullet\blacktriangle\}$, станции $s_4 - \{ \blacksquare\blacksquare\blacksquare\blacktriangle\}$. Совокупность точек r -го типа в j -ой ячейке принимается компактным множеством E_j . На рис. 1 множества E_j выделены в тех ячейках сетки, которые содержат точки, обозначенные символом (●). Номера j в таких ячейках проставлены в нижнем правом углу, $j = 1, 2, \dots, 6$.

Введенное представление точек множеств E_r совокупностью компактных множеств E_j позволяет записать задачу распределения точек по станциям в следующем виде:

$$\begin{aligned} \text{Задача } T^* : \quad \min L &= \sum_{v=1}^V \sum_{j=1}^m c_{vj} x_{vj} ; \\ \sum_{j=1}^m x_{vj} &= a_v, \quad v = 1, 2, \dots, V ; \\ \sum_{v=1}^V x_{vj} &= b_j, \quad j = 1, 2, \dots, m . \end{aligned}$$

Здесь x_{vj} – число точек множества E_j , подключаемых к станции v ;

c_{vj} – расстояние между станцией v и центром множества E_j ;

a_v – число мест подключения точек r -го вида в станции v ;

b_j – число точек r -го вида, составляющих множество E_j .

Задача T^* полностью соответствует классической форме записи задачи транспортного типа. Условие равенства объемов производства и объемов потребления здесь также соблюдается: $\sum_{v=1}^V a_v = \sum_{j=1}^m b_j$.

На рис. 2 приведена схема «перевозок» транспортной задачи T^* , построенная для примера ТП на рис. 1.

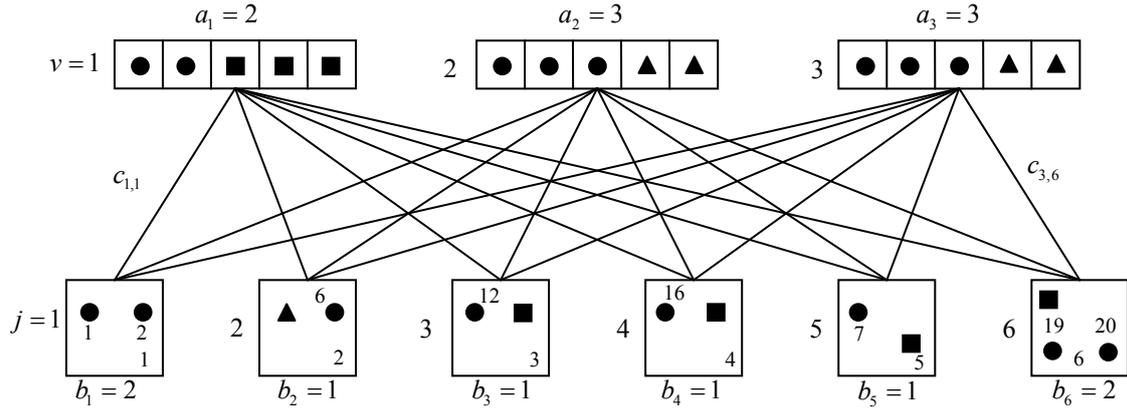


Рис. 2. Пример схемы «перевозок» для задачи T^*

Пункты производства с объемами a_v соответствуют трем станциям, содержащим точки типа (\bullet) ($a_1 = 2, a_2 = 3, a_3 = 3$), а пункты потребления с объемами b_j шести ячейкам сетки ТП, содержащим точки типа (\bullet) .

Общее оптимальное решение задачи распределения точек множества E складывается из решений задач T^* для множеств E_r . Заметим также, что в диссертации данная задача формулируется и для условий, когда сетка на ТП не накладывается. Получаемая при этом задача T , отличается от T^* тем, что в ней пунктами потребления являются точки множества E_r и, соответственно, величины $b_j = 1$.

В последнем разделе второй главы вводится понятие компактного разбиения и показано, что задача размещения станций и распределения по ним точек является задачей получения компактных разбиений. Рассматривается ситуация, когда точки на ТП одного типа, а станции подключают одинаковое число точек.

Пусть на ТП задано множество точек $Q = \{e_i\}, i = 1, 2, \dots, n$. Разобьем подмножество Q на совокупность множеств $Q_j, j = 1, 2, \dots, m$, так, что:

$$Q_j \neq \emptyset; |Q_j| = v_j; 1 \leq v_j \leq v, \cup_{j=1}^m Q_j = Q; Q_j \cap Q_k = \emptyset; j, k = 1, 2, \dots, m. \quad (10)$$

Будем рассматривать разбиения, получаемые для условия $v_j = v$, то есть $vm = n$. Выражение (10) в этом случае задает совокупность W разбиений, которые можно сравнивать между собой. Введем оценку R_w разбиения $w \in W$:

$$R_w = \sum_{j=1}^m R_j, R_j = \sum_{e_i \in Q_j} d(e_i, e_j^\circ), \text{ где } d(e_i, e_j^\circ) \text{ – расстояние от точки } e_i \text{ до центра } e_j^\circ$$

множества Q_j .

Разбиение $\{Q_j^*\}_w$ назовем компактным, если среди всех разбиений W ему соответствует величина $R_w^* = \min_{w \in W} R_w$. Используются также относительные оцен-

$$\text{ки: } \rho_w = \sum_{j=1}^m \rho_j, \quad \rho_j = R_j / v_j, \quad \rho_w^* = \min_{w \in W} \rho_w.$$

В ряде случаев вместо оценок R удобнее пользоваться оценками L , которые определяются суммами расстояний между точками в множествах Q_j :

$$L_w = \sum_{j=1}^m L_j, \quad L_w = \sum_{(e_i, e_t) \in Q_j} d(e_i, e_t), \quad L_w^* = \min_{w \in W} L_w.$$

В относительных величинах данные оценки имеют вид:

$$\xi_w = \sum_{j=1}^m \xi_j, \quad \xi_j = L_j / v_j, \quad \xi_w^* = \min_{w \in W} \xi_w.$$

В работе вводятся еще две оценки, одна из которых направлена на получение разбиений $\{Q_j\}_w$ с минимальной величиной наибольшего расстояния от точек множества Q_j до его центра. Вторая оценка используется при разбиении взвешенного графа с множеством вершин Q_j по критерию максимальной суммы весов ребер подграфов.

Более глубоко исследованы оценки R и L . Экспериментально была показана возможность их совместного применения и взаимозаменяемость. Так на рис. 3, показана зависимость оценок R и L от параметра m , построенная для разбиений на множестве Q с $n = 30$. Из рис. 3 следует, что разбиение, выбранное как лучшее по оценке L , является лучшим и по оценке R .

Для получения разбиения $w \in W$ по критерию минимизации оценки R_w в работе предложен оригинальный алгоритм, который включает следующие операции. По алгоритму α_1 формируется m полюсов. Полюса принимаются центрами и относительно них решается задача T , распределяющая точки множества Q по данным центрам. В результате получается исходное разбиение множества Q на множества Q_j , $j = 1, 2, \dots, m$. В каждом множестве Q_j определяется новый центр, производится перенос центра из бывшего в новый и решается задача T относительно новых центров. Если в результате решения задачи T происходит перераспределение точек между множествами Q_j , то в них вновь определяются центры и решение задачи T повторяется. Процесс переноса центров и решения задачи T повторяется пока очередное разбиение не совпадет с предыдущим. В этом случае перенос центров невозможен. Полученное устойчивое разбиение именуется локальным компактным разбиением (ЛК – разбиением).

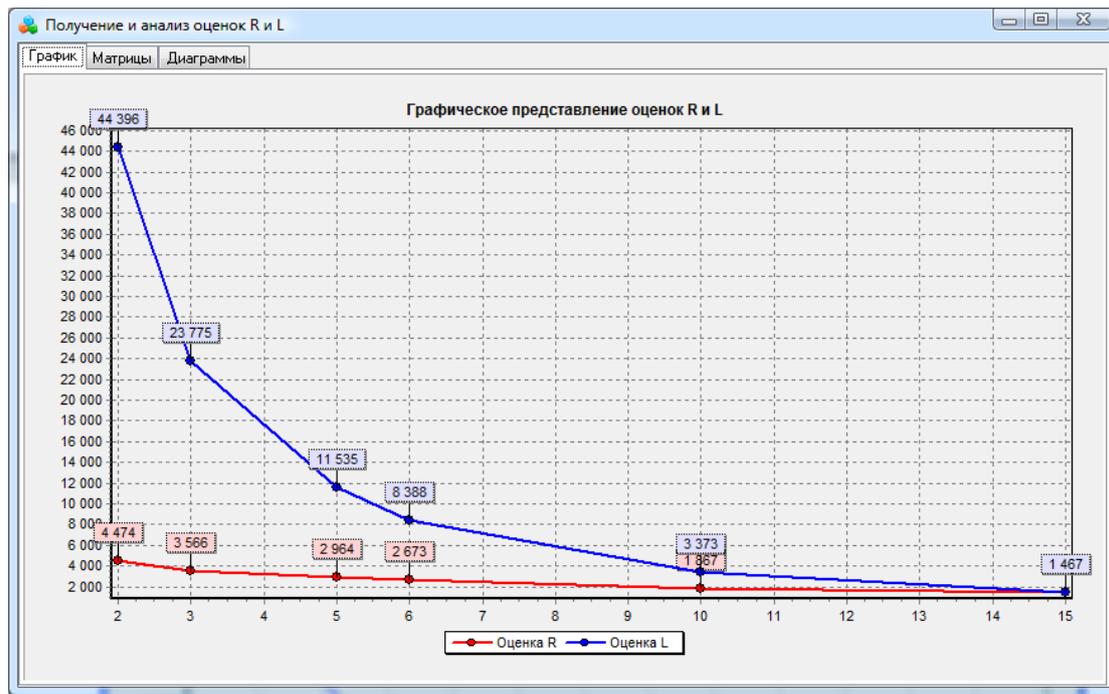


Рис. 3. Пример сопоставления оценок R_w и L_w

Экспериментальные исследования показали, что к одному и тому же ЛК – разбиению сходятся многие исходные варианты разбиений. При этом даже значительный разброс в оценках компактности исходных разбиений быстро «сглаживается» после решения задачи T . Замечено также, что подобрать исходные разбиения для получения разных ЛК – разбиений достаточно трудно. Из этого следует важный вывод о том, что качество разбиения получаемого с помощью данного алгоритма, не столь критично к исходному варианту разбиения. Этот вывод применим и к задаче размещения станций и распределения по ним точек.

Качество ЛК – разбиения предложено оценивать относительно некоторого «идеального» разбиения. Вводится понятие «идеального» разбиения и дается алгоритм вычисления для него оценки компактности R_0 . Оценка R_0 принимается в качестве граничной оценки компактности. По ней можно судить о степени приближения полученного ЛК – разбиения к компактному.

В третьей главе основное внимание уделено решению задачи построения топологии сети вычислительной системы при заданном размещении станций на ТП. Метод решения задачи построения топологии сети заключается в выполнении последовательности операций: построение информационного графа; определение плана использования ресурсов и построение графа передач данных между станциями сети; выбор базовой сети и анализ конфликтов в магистральных сети при передаче данных; выбор варианта подключения станций к магистралям сети.

Информационный граф задает объемы передач данных между вершинами ГПД за один цикл моделирования и строится на основе ГПД и условий функ-

ционирования объекта управления. На рис. 4 приведен пример информационного графа с параметрами, полученными для цикла моделирования в 10 тактов.

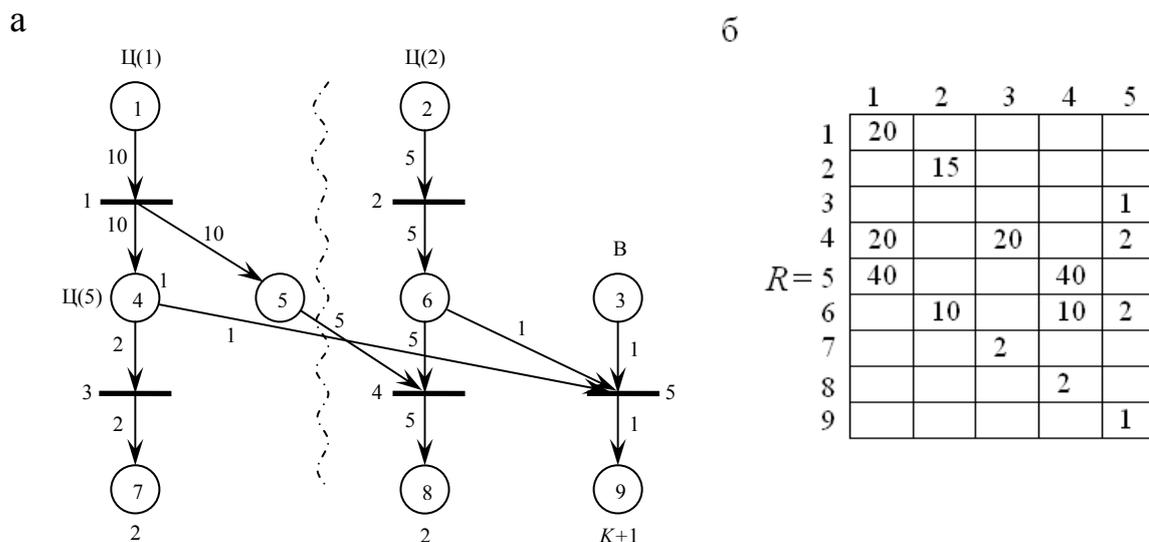


Рис. 4. Пример информационного графа и матрицы R

Данные d_q пронумерованы в кружках, а номера модулей f_m проставлены рядом с планками. Число на дуге обозначает частоту передаваемых по ней данных за один цикл моделирования. Рядом с входными данными указаны условия их поступления. Для выходных данных указывается число тактов цикла обновления. На основе указанных параметров для каждого ребра графа определяется вес r_{qm} – объем данных, передаваемых между вершинами d_q и f_m за один цикл моделирования, $r_{qm} = p_q \rho_m c_{qm}$. Здесь p_q – размер памяти, занимаемый данным d_q ; ρ_m – частота выполнения модуля f_m в цикле моделирования; c_{qm} – число состояний данных d_q , передаваемых между вершинами d_q и f_m в соответствии с условиями селекции. Матрица $R = \|r_{qm}\|$ на рис. 4б получена для значений $\rho_m \Rightarrow \{10, 5, 2, 5, 1\}$ и $p_q \Rightarrow \{2, 3, 1, 2, 4, 2, 1, 1, 1\}$, $c_{4,3} = 5$, $c_{5,4} = 2$, для остальных дуг $c_{qm} = 1$.

План использования ресурсов определяется путем разрезания информационного графа на минимально связанные части. Число частей равно числу станций. Модули и данные в каждой части по сумме требуемых ресурсов не должны превышать возможности станции. На основе разрезания строится граф передач данных, вершинами в котором являются станции, а веса ребер обозначают объемы данных, передаваемых между станциями за один цикл моделирования.

Пример графа передач данных для шести станций, построенный для информационного графа, содержащего 26 вершин, показан на рис. 5. Общий объем данных, передаваемых в сети r_w , для варианта разрезания $w \in W$, в данном примере составляет 155 единиц. Если пропускную способность магистрали сети обозначить величиной φ , определяющей объем данных, передаваемый за один такт моделирования, то для успешной работы локальной сети на базе одной магистрали, должно выполняться условие $r_w / \varphi \leq \mu t_k^*$. Здесь μ – число максимальных циклов обновления выходных данных t_k^* , принятое при задании цикла моделирования.

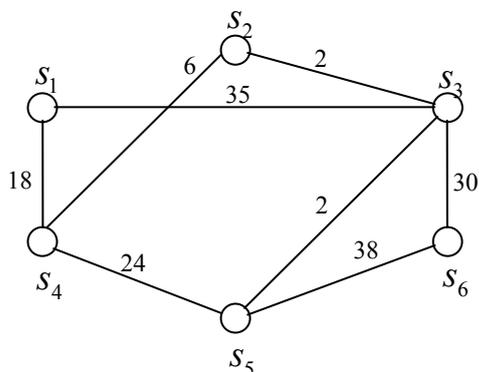
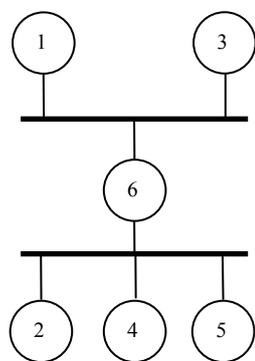


Рис. 5. Пример графа передачи данных

Минимально необходимое число магистралей в сети определяется отношением $(r_w / \varphi) / \mu t_k^*$ с учетом округления результата в большую сторону. Так, для нашего примера, при $r_w = 155$, $\varphi = 10$, $\mu = 1$, $t_k^* = 12$ имеем $(155/10)/12 \cong 1,3$. Таким образом, в сети должно быть не менее двух магистралей. Из библиотеки базовых сетей выбирается вариант сети, построенной на двух магистралях. Сеть и вариант подключения к ней шести станций показаны на рис. 6а. На рис. 6б приведена матрица наличия конфликтов $Q = \|q_{vk}\|$, полученная для данной сети и графа передач данных (рис. 5).

Число строк и столбцов матрицы равно числу ребер графа. Ребра обозначены кодовыми номерами составленными из номеров станций, связываемых ребром. Элемент $q_{vk} = 1$, если пары станций ребер v и k при передаче данных в сети имеют конфликт по доступу к магистрали, $q_{vk} = 0$, в противном случае. В матрице Q выделены строки и столбцы, у которых все элементы $q_{vk} = 1$, $v \neq k$.

а



б

	13	14	23	24	35	36	45	56
13		1	1		1	1		
14	1		1	1	1	1	1	1
23	1	1		1	1	1	1	1
24		1	1		1		1	1
35	1	1	1	1		1	1	1
36	1	1	1		1			
45		1	1	1	1			1
56		1	1	1	1		1	

Рис. 6: а – вариант подключения станций к базовой сети; б – матрица наличия конфликтов Q

Диаграмма совмещения передач данных для принятого варианта подключения станций к магистралям сети приведена на рис. 7б. Для удобства построения диаграммы матрица Q представлена графом (рис. 7а) Из диаграммы следует, что общая загрузка сети с учетом совмещения передач данных составляет 90 единиц или $90/10=9$ тактов, что укладывается в цикл моделирования $t_k^*=12$ тактов.

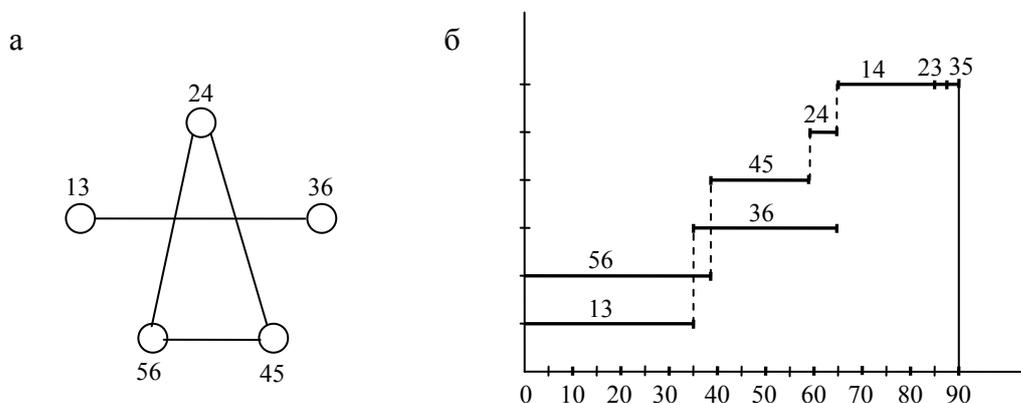


Рис. 7: а – граф матрицы Q ; б – диаграмма совмещения передач

Полученный результат следует рассматривать как один из возможных для выбранной базовой сети и варианта подключения станций. Наилучшее решение получается в результате перебора вариантов. Потребность в сокращении переборов при поиске наилучшего решения появляются, когда число станций приближается к 20. В этом случае сокращение переборов достигается путем разрезания графа передач данных на подграфы. В этом случае станции, соответствующие вершинам подграфа, подключаются к одной из магистралей сети.

В данной главе рассматриваются также вопросы построения модели алгоритма функционирования объекта и согласования ее с моделью программной нагрузки с учетом полученной топологии сети вычислительной системы. Предлагается также методика получения последовательности обхода станций сети при разработке планов их обслуживания.

В четвертой главе изложены результаты программирования и экспериментальной проверки предлагаемых в диссертации алгоритмов решения основных задач построения архитектуры и топологии сети вычислительной системы. Для программирования были выбраны те задачи, алгоритмы решения которых содержат эвристические правила, требующие уточнения в ходе экспериментальных проверок. Экспериментальные исследования проводились на специально разработанных контрольных примерах, которые позволили более полно имитировать исходные условия задачи и выявлять недостатки в работе алгоритмов.

Показано место предлагаемых алгоритмов в составе подсистемы проектирования архитектуры и топологии сети территориально распределенной вычислительной системы. Данные алгоритмы в составе подсистемы представлены четырьмя программными средствами: «Полюс» – формирует полюса, множества точек и центров для размещения станций; «Топология» – оптимизирует рас-

пределение точек по станциям и определяет локальное компактное разбиение; «Сеть» – выбирает базовую сеть и вариант подключения станций к магистралям сети; «Редактор архитектур» – описывает модель архитектуры вычислительной системы.

Разработка программных средств выполнена в среде Borland Delphi 7.0 для операционной системы Windows XP. Суммарный объем исходного текста программ составил более 6000 тысяч строк. Описание программных средств сопровождается окнами с результатами их работы. Подробно изложено применение программных средств для анализа наземной метеорологической наблюдательной сети Росгидромета. Приведены результаты решения задачи выбора первичных и вторичных каналов связи в данной сети.

Основные результаты работы

1. Предложен состав задач и разработаны алгоритмы и программные средства их решения, которые на основе сведений о территории размещения оборудования объекта управления и условий его функционирования позволяют аналитическим путем определить архитектуру и топологию сети вычислительной системы, при проектировании СРВ по управлению данным объектом. Предлагаемые алгоритмы позволяют преодолеть недостатки известных подходов к решению подобных задач, являются наглядными в составе интерактивного сценария проектирования, эффективными и способными решать задачи, размерность которых и достигаемая при этом точность приемлемы для практического применения.

2. Задача определения минимального числа станций, способных подключить все терминальные точки топологического поля, сформулирована как целочисленная задача линейного математического программирования, известная как задача покрытия. Предложена методика анализа возможности выполнения программной нагрузки на полученной совокупности станций с учетом ограничений реального времени.

3. Предложен метод решения задачи размещения станций на топологическом поле объекта управления, основанный на разбиении множества терминальных точек на расширенную совокупность компактных подмножеств и выбора среди них предпочтительных для размещения станций. Для выбора предпочтительных подмножеств разработан алгоритм по схеме метода ветвей и границ.

4. Задача распределения терминальных точек по станциям сформулирована как задача математического программирования транспортного типа. Введены оценки компактности для отдельных множеств, полученных в результате решения транспортной задачи и разбиения множества терминальных точек на подмножества в целом. Исследованы некоторые свойства оценок компактности.

5. Предложен алгоритм получения разбиений с локальным минимумом оценки компактности (ЛК – разбиений), в основу которого положена итерационная процедура применения транспортной задачи и анализа оценки разбиения. Данная процедура является эффективным инструментом, который мо-

жет быть использован в других приложениях при поиске разбиений, компактных относительно принятой оценки.

6. Разработан метод выбора топологии сети по критерию минимальных суммарных затрат времени на передачу данных. Метод включает отображение условий функционирования объекта управления на модель программной нагрузки и построение информационного графа, распределение программной нагрузки по станциям и определение объема данных, передаваемых между станциями, построение диаграммы совмещения параллельных передач данных. Задача распределения программной нагрузки по станциям сформулирована как задача разрезания информационного графа на минимально связанные подграфы.

7. Выполнена алгоритмизация и программная реализация методов построения архитектуры и топологии сети многопроцессорной вычислительной системы для управления объектами с территориально распределенным оборудованием. Программные средства разработаны в составе четырех программ: «Полюс», «Топология», «Сеть», «Редактор архитектур». Результаты диссертационных исследований внедрены в ОАО «Омский институт системотехники» при создании автоматизированной системы оперативно-диспетчерского управления тепловодоресурсами (АСОДУЭ-ТВР) для предприятий электрических и тепловых сетей (Норильский промышленный район), в ГУ «Всероссийский НИИ гидрометеоинформации – мировой центр данных» для анализа наземной наблюдательной сети рамках ФЦП «Мировой океан» для Северо-Кавказского Управления Гидрометеослужбы (УГМС), а также Камчатского и Сахалинского УГМС в рамках ФЦП «Цунами», в учебном процессе ТПУ при изучении дисциплины «Автоматизированное проектирование распределенных СРВ».

Основные публикации по теме диссертации

1. Погребной А.В. Оптимизация топологии компонентов вычислительной системы при проектировании систем реального времени. В кн.: Математическое и программное обеспечение проектирования систем. Вып.2. Томск, ТПУ, 2002, с53-55.

2. Погребной А.В. Построение модели для топологически распределенной динамической системы. – Кибернетика и вуз, вып. 30 , Томск, Изд-во ТПУ, 2003. – С. 82–86.

3. V.K.Pogrebnoy, A.V.Pogrebnoy. Efficient placement of stations of topologically distributed multiprocessing computing systems. In. Proc. 8th Korean-Russia International Symposium on Science and Technology (KORUS-2004), Vol.. 1, pp.137-141, 2004.

Погребной В.К., Погребной А.В. О рациональном размещении станций топологически распределенных многопроцессорных вычислительных систем.

4. Погребной А.В. Задача определения числа станций многопроцессорных систем управления. Доклады II Всероссийской научно-практической конференции студентов «Молодежь и современные информационные технологии». Февраль 2004 г. Томск: изд-во ТПУ, 2004.

5. Pogrebnoy A.V. The solution of a multiplanimetric problem of traveling salesman at the analysis of territorially distributed technical system. Proceedings of the 9 Russian–Korean international symposium on science and technology.— Novosibirsk, 2005.

Погребной А.В. Решение многоконтурной задачи коммивояжера при анализе территориально распределенной технической системы.

6. Погребной А.В. Выбор архитектуры локальной сети при проектировании распределенной системы реального времени.//Современные техника и технологии: Труды IX Международной научно – практической конференции молодых ученых. – Томск, 2006. – Т.2.

7. Погребной А.В. Определение числа и топологии размещения станций многопроцессорной вычислительной системы.// Известия Томского политехнического университета. – 2006. – Т.309.№7 – С.160–164.

8. Погребной А.В. Задача топологической децентрализации в территориально распределенных технических системах.//Современные техника и технологии: Труды X Международной научно – практической конференции молодых ученых. – Томск, 2007. – Т.2.

9. Погребной А.В. Определение объемов передач данных в сети вычислительной системы для заданной модели программной нагрузки.// Известия Томского политехнического университета. – 2007. – Т.310.№3 – С.103–107.

10. Погребной А.В, Погребной Д.В. Проектирование структуры локальной сети для распределенной вычислительной системы реального времени.// Известия Томского политехнического университета. – 2007. – Т.310.№5 – С.97–101.