Хоботов Е.Н.

Институт системного анализа РАН, Москва.

О ПОСТРОЕНИИ СИСТЕМ РАСПРЕДЕЛЕННОГО МОДЕЛИРОВАНИЯ НЕКОТОРЫХ ТИПОВ

Рассматриваются принципы и методы организации взаимодействия моделей, описывающих различные подсистемы некоторой системы, а также способы организации процессов моделирования сложных систем для изучения их свойств с использованием уже существующих моделей подсистем изучаемых систем. Исследуются алгоритмы, обеспечивающие взаимодействие таких моделей.

1. Введение

В последние годы большой интерес вызывают задачи, связанные с моделированием сложных систем, которые в свою очередь могут состоять из достаточно большого количества различных подсистем. Для моделирования таких систем используются различные методы, важнейшее место среди которых занимают методы имитационного моделирования [1-5], а также строятся модели, расчеты которых часто сводятся к решению задач линейного и нелинейного программирования [6-9]. Во многих случаях, однако, для адекватного описания реальных систем приходиться учитывать большое количество различных параметров, что приводит к необходимости решать задачи большой размерности. Это значительно затрудняет процесс их моделирования. Кроме того, очень часто при моделировании сложных систем, в состав которых входит несколько подсистем, возникают ситуации, когда для моделирования отдельных подсистем такой системы уже разработаны модели и имеется реализующее их программное обеспечение. В связи с этим возникает желание использовать имеющееся программное обеспечение, особенно доказавшее свою эффективность, также и при моделировании всей системы.

Для решения такой проблемы в работе [9] была предложена идея построения систем распределенного моделирования. В соответствии с этой идеей основную модель для исследования какой-либо системы целесообразно строить с использованием разработанных ранее моделей ее подсистем и

реализующего их программного обеспечения. Идея построения систем распределенного моделирования рассматривалась на примере решения задачи, в которой требовалось связать в единый комплекс N математических моделей, каждая из которых формулировалась как задача математического программирования. Для организации взаимодействия таких моделей предлагалось использовать метод гладких штрафных функций с введением в каждую модель по определенным правилам специальных параметров.

В работе [10] был рассмотрен случай, когда исследуемые системы состоят из подсистем, которые описываются линейными моделями. В таких случаях процесс взаимодействия моделей может быть существенно упрощен. Однако в работе не было приведено доказательство сходимости алгоритма, который обеспечивает получение решения для задачи, возникающей при исследовании таких систем.

В данной работе рассматривается метод декомпозиции задач, для решения которых непосредственное использование схемы таких известных методов декомпозиции, как схемы Данцига-Вулфа и Корнаи-Липтака [3-4] оказывается неэффективным. Кроме того, в работе рассматриваются прицины и методы организации взаимодействия отдельных моделей, которые могут иметь нелинейные ограничения, и описываются различные подсистемы изучаемой системы. Приводится доказательство сходимости предлагаемого метода.

2. Постановка задачи

Процесс организации взаимодействия отдельных моделей такого типа идейно весьма близок к процессу декомпозиции сложной задачи большой размерности на ряд задач существенно меньшей размерности. Действительно при декомпозиции задачи большой размерности ее стремятся разделить на ряд задач существенно меньшей размерности и организовать процесс их решения так, чтобы на основе их решений получить решение исходной задачи.

Очень часто возникают и обратные задачи к задачам декомпозиции, когда появляется потребность в изучении свойств и поведения некоторой сложной системы, для которой уже существуют модели, описывающие функционирование ее отдельных подсистем, или же построение отдельных моделей подсистем оказывается проще, чем построение общей модели всей системы. В таких случаях возникает проблема, как с небольшими затратами средств и времени построить достаточно адекватную модель для исследования свойств

системы, включающей подсистемы, для которых уже существуют модели, позволяющие исследовать эти подсистемы.

Важным вопросом также является вопрос о возможности использования моделей отдельных подсистем для синтеза системы с требуемыми характеристиками из таких подсистем. Подобные задачи возникают при проектировании сложных систем, которые часто создаются на базе уже разработанных подсистем.

В ряде случаев возникает еще один вопрос - можно ли и как при программной реализации модели исследуемой системы использовать ранее разработанное программное обеспечение, реализующее модели отдельных подсистем, входящих в состав такой системы?

Положительное решение этих вопросов может значительно сократить средства и время на исследование свойств и характеристик уже существующих систем или на проектирование новых систем с заданными характеристиками.

Для изучения этих задач и получения ответов на поставленные вопросы рассмотрим возможность декомпозиции, а затем и организации взаимодействия отдельных модулей на примере задачи выпуклого программирования следующего вида:

$$J = \min\{\sum_{\mathrm{k}=1}^{N}\sum_{i=1}^{n_k} c_i^k x_i^k + \sum_{k=1}^{N}\sum_{i=1}^{m_k} \bar{c}_i^k y_i^k\}, \tag{1}$$

$$\sum_{k=1}^{N}\sum_{i=1}^{m_k} a_{ji}^k y_i^k \le \breve{b}_j, \quad j = 1,...,\overline{m}, \tag{2}$$

$$\sum_{i=1}^{n_k} b_{li}^k x_i^k + \sum_{j=1}^{m_k} \tilde{b}_{li}^k y_j^k \le \hat{b}_l^k, \quad l = 1,...,\overline{n}_k, \quad k = 1,...,N, \tag{3}$$

$$f_i^k(x^k, y^k) \le \widetilde{b}_i^k, \quad i = 1,...,\widetilde{m}_k, \quad k = 1,...,N, \tag{4}$$

$$0 \le x_i^k \le M_i^k, \quad i = 1,...,n_k, \quad 0 \le y_j^k \le \widetilde{M}_j^k, \quad j = 1,...,m_k,$$

где x_i^k, ($i = 1,...,n_k$, $k = 1,...,N$), y_i^k ($i = 1,...,m_k$, $k = 1,...,N$) – переменные; c_i^k, $\bar{c}_i^k$, a_{ji}^k, $\breve{b}_j$, $\tilde{b}_{li}^k$, b_{li}^k, $\hat{b}_l$ и $\widetilde{b}_i$ – постоянные коэффициенты; $f_i^k(x^k, y^k)$ – выпуклые функции переменных $x_1^k,...,x_{n_k}^k, y_1^k,...,y_{m_k}^k$; а M_i^k,

$\widetilde{M}_j^k$ – достаточно большие постоянные величины.

Очень часто, особенно при моделировании производственных систем [11] с использованием моделей вида (1)-(4) размерность «связывающих» переменных y_j^k оказывается малой, а размерность переменных x_i^k достаточно большой. Обычно переменные x_i^k используются для описания структуры отдельных элементов и подсистем исследуемой системы или процессов, происходящих в них, а переменные y_j^k используются для описания связей между элементами и подсистемами.

Такая ситуация возникает при моделировании систем, когда в их состав входит несколько подсистем и такие подсистемы являются достаточно сложными, а связи между ними не такими тесными, т. е. могут быть описаны переменными существенно меньшей размерности.

Для решения задачи (1)-(4) возникает идея декомпозировать ее на отдельные подзадачи. С одной стороны декомпозиция этой задачи на отдельные подзадачи позволит снизить размерность каждой из решаемых задач, а с другой стороны может подсказать метод организации совместной работы различных ранее независимых моделей.

Здесь, однако, следует отметить, что, как видно из структуры задачи (1)-(4), для ее декомпозиции непосредственное использование схемы таких известных методов декомпозиции, как схемы Данцига-Вулфа или Корнаи-Липтака [3-4], оказывается неэффективным.

3. Принципы построения систем распределенного моделирования

Рассмотрим схему декомпозиции, в которой задача (1)-(4) разделяется на координирующую задачу минимизации функционала:

$$\tilde{J} = \min\{\sum_{k=1}^{N}\beta_k\sum_{i=1}^{m_k}\hat{c}_i^k y_i^k + \alpha_v t_v\}, \tag{5}$$

при наличии следующих ограничений:

$$\sum_{k=1}^{N}\sum_{i=1}^{m_k} a_{ji}^k y_i^k \le \breve{b}_j, \quad j=1,...,\overline{m}, \quad 0 \le y_i^k \le \widetilde{M}_i^k, \quad i=1,...,m_k, \tag{6}$$

$$y_i^k - z_i^k + u_i^k - w_i^k = 0, \quad i=1,...m_k, \quad k=1,...,N, \tag{7}$$

$$\sum_{k=1}^{N}\sum_{i=1}^{m_k}\{u_i^k + w_i^k\} \le t_v, \quad t_v \ge 0, \quad u_i^k \ge 0, \quad w_i^k \ge 0, \tag{8}$$

и на k ($k = 1,...,N$) отдельных задач минимизации функционалов:

$$\hat{J}_k = \min\{\sum_{i=1}^{n_k} c_i^k x_i^k + (1-\beta_k)\sum_{i=1}^{m_k}\hat{c}_i^k z_i^k + \tilde{\alpha}_v^k \tilde{t}_v^k\}, \tag{9}$$

при наличии следующих ограничений:

$$\sum_{i=1}^{n_k} b_{li}^k x_i^k + \sum_{i=1}^{m_k}\tilde{b}_{li}^k z_i^k \le \hat{b}_l^k, \quad l = 1,...,\overline{n}_k, \tag{10}$$

$$f_i^k(x^k, z^k) \le \tilde{b}_i^k, \quad i = 1,...,\tilde{m}_k, \tag{11}$$

$$0 \le x_l^k \le M_l^k, \quad l = 1,...,n_k \qquad 0 \le z_j^k \le \tilde{M}_j^k, \quad j = 1,...,m_k, \tag{12}$$

$$y_i^k - z_i^k + \tilde{u}_i^k - \tilde{w}_i^k = 0, \quad i = 1,...m_k, \tag{13}$$

$$\sum_{i=1}^{m_k}\{\tilde{u}_i^k + \tilde{w}_i^k\} \le \tilde{t}_v^k, \quad \tilde{t}_v^k \ge 0, \quad \tilde{u}_i^k \ge 0, \quad \tilde{w}_i^k \ge 0. \tag{13}$$

Здесь β_k – постоянный параметр ($0 \le \beta_k \le 1$), который не меняется в процессе решения, z_i^k – постоянный параметр в задаче (5)-(8) и переменная в задаче (9)-(13), y_i^k – постоянный параметр в задаче (9)-(13) и переменная в задаче (5)-(8), α_v и $\tilde{\alpha}_v^k$ – штрафные коэффициенты, величина которых увеличивается после выполнения каждой итерации v метода декомпозиции, u_i^k, w_i^k, u_i^k, w_i^k, t_v и $\tilde{t}_v$ – вспомогательные переменные. Остальные обозначения определены выше.

Увеличение штрафных коэффициентов α_v и $\tilde{\alpha}_v^k$ будет «способствовать» уменьшению разности

$$\sum_{k=1}^{N}\sum_{i=1}^{m_k}\left|y_i^k - z_i^k\right|$$

и достижению условий $y_i^k = z_i^k$ для всех i и k, которые обеспечивают получение решения задачи (1)-(4) при $v \to \infty$.

Если параметр β_k положить равным нулю, то в координирующей задаче останутся только вспомогательные переменные и переменные, влияющие на

распределение общих ресурсов, а в задачах (9)-(13) – вспомогательные переменные и переменные, описывающие функционирование отдельных подсистем исследуемой системы. Поэтому такое значение параметра β_k часто оказывается наиболее предпочтительным.

При построении схем и методов декомпозиции необходимо обеспечить, чтобы задачи, на которые декомпозируется исходная задача, всегда имели решение, если исходная задача имеет решение. Поэтому алгоритмы, предназначенные для получения согласованного решения задач (5)-(8) и (9)-(13) должны строиться с учетом этого требования.

Из описания этих задач также видно, что величина функционала задачи (1)-(4) будет совпадать с величиной суммы:

$$\sum_{k=1}^{N}\beta_k\sum_{j=1}^{m_k}\hat{c}_j^k y_j^k + \sum_{k=1}^{N}\{\sum_{i=1}^{n_k}c_i^k x_i^k + (1-\beta_k)\sum_{j=1}^{m_k}\hat{c}_j^k z_j^k\},$$

в которой первое слагаемое входит в функционал задачи (5)-(8), а второе и третье - в функционалы задач (9)-(13) при условии, что $y_i^k = z_i^k$ при всех i и k.

В связи с этим для решения задачи (1)-(4) может использоваться схема декомпозиции, в которой последовательно решаются задача (5)-(8) и задачи (9)-(13) при $k = 1,...,N$. Однако для достижения равенства значений переменных y_i^k и z_i^k в такой схеме декомпозиции могут использоваться различные алгоритмы.

Рассмотрим один из таких алгоритмов, который позволяет путем последовательного решения задачи (5)-(8) и задач (9)-(13) получить решение исходной задачи (1)-(4), а также условия его сходимости.

4. Алгоритмы согласования работы отдельных моделей

По шагам указанный алгоритм описывается следующим образом.

Шаг 1. Решаются задачи (5)-(6) и (9)-(11) соответственно при $\alpha_0 = 0$ и $\tilde{\alpha}_0^k = 0$. Если хотя бы одна из этих задач не имеет решения, то и задача (1)-(4) также не имеет решения.

Шаг 2. Решаются задачи (5)-(8) и (9)-(13) при $\alpha_1 = 1$ и $\tilde{\alpha}_1^k = 1$. Причем для задачи (4)-(7) в качестве параметров z_i^k выбираются вектора z_i^k, определенные в результате решения задач (9)-(13) при $k = 1,...,N$, а для задач (9)-(13) в качестве параметров y_i^k выбираются величины y_i^k, определенные

в результате решения задачи (5)-(8). Такой выбор параметров z_i^k и y_i^k соответственно для задач (5)-(8) и (9)-(13) будет осуществляться в процессе всего решения и дополнительно оговариваться дальше не будет. Для каждой из этих задач определяются величины t_v и $\widetilde{t}_v$ ($v=1$), где v - номер итерации метода.

Шаг 3. Если величина вычисленной суммы ($t_v + \sum_{k=1}^{N} \widetilde{t}_v^{\,k}$) окажется меньше $\bar{\varepsilon}$, где $\bar{\varepsilon}$ достаточно малая положительная величина, которая определяет точность получаемого решения, то процесс вычислений прекращается. В противном случае производится переход к шагу 4.

Шаг 4. Решается задача (5)-(8) со штрафным коэффициентом $\alpha_{v+1} = \hat{M}\alpha_v$, где $\hat{M}$ - положительная величина строго большая 1. При этом значении α_{v+1} определяется величина t_{v+1} и следует переход к шагу 5.

Шаг 5. Решаются задачи (9)-(13) со штрафными коэффициентами $\widetilde{\alpha}_{v+1}^k = \breve{M}\widetilde{\alpha}_v^k$, где $\breve{M}$ - положительная величина строго большая 1. Введение штрафных коэффициентов α_v и $\widetilde{\alpha}_v^k$ позволяет ускорить сходимость метода. При этих значениях $\widetilde{\alpha}_{v+1}^k$ определяются величины $\widetilde{t}_{v+1}^k$ ($k=1,...,N$) и следует переход к шагу 6.

Шаг 6. Вычисляется сумма $\left(t_{v+1} + \sum_{k=1}^{N} \widetilde{t}_{v+1}^{\,k}\right)$. Если величина этой суммы окажется меньше $\bar{\varepsilon}$, где $\bar{\varepsilon}$ достаточно малая положительная величина, которая определяет точность получаемого решения, то процесс вычислений прекращается. В противном случае производится проверка выполнения неравенства

$$\left(t_{v+1} + \sum_{k=1}^{N} \widetilde{t}_{v+1}^{\,k}\right) \le \varepsilon \left(t_v + \sum_{k=1}^{N} \widetilde{t}_v^{\,k}\right) \quad , \tag{15}$$

где положительная величина $\varepsilon < 1$, ($\varepsilon \approx 0.0 \div 0.9$). Если это неравенство выполняется, то следует переход на шаг 4.

В противном случае в модель (4)-(8) включается дополнительное ограничение $t_{v+1} \le \varepsilon\, t_v$, а в каждую k-ю модель (9)-(13) – ограничение $\widetilde{t}_{v+1}^k \le \varepsilon\, \widetilde{t}_v^{\,k}$. В этих ограничениях t_v и $\widetilde{t}_v^{\,k}$ являются параметрами, значения которых равны значениям соответствующих переменных, полученных на предыдущей итерации метода. Если задачи с такими ограничениями окажутся несовместными, то и задача (1)-(4) будет несовместной.

Рассмотрим теперь условия, при которых такой алгоритм сходится, т. е. позволяет получить решение исходной задачи.

Теорема 1.

Пусть задача (1)-(4) имеет не пустое множество решений, все функции $f_i^k(x^k, y^k)$ ($i = 1,..., \widetilde{m}_k$) выпуклы по переменным $x_1^k,..., x_{n_k}^k, y_1^k,..., y_{m_k}^k$ ($k = 1,..., N$). Тогда алгоритм, вычислительная схема которого приведена выше, позволяет определить одно из решений этой задачи при $v \to \infty$ и $\alpha_v, \widetilde{\alpha}_v^k \to \infty$.

Доказательство.

Из структуры задач (5)-(8) и (9)-(13), как уже отмечалось выше, видно, что величина функционала (1) задачи (1)-(4) будет равна величине суммы:

$$\sum_{k=1}^{N} \beta_k \sum_{j=1}^{m_k} \hat{c}_j^k y_j^k + \sum_{k=1}^{N} \left(\sum_{i=1}^{n_k} c_i^k x_i^k + (1 - \beta_k) \sum_{j=1}^{m_k} \hat{c}_j^k z_j^k \right),$$

в которой первое слагаемое входит в функционал (5) задачи (5)-(8), а второе и третье - в функционалы (9) задач (9)-(13) при условии, что $y_i^k = z_i^k$ при всех i и k.

Кроме того, когда при решении задач (5)-(8) и (9)-(13) будут выполнены условия $y_i^k = z_i^k$ для всех i и k, то и ограничения (2)-(4) для переменных x_i^k и y_i^k будут удовлетворены при максимальном значении функционалов (5) и (9), что обеспечит и максимальное значение функционала (1). Таким образом, при равенстве $y_i^k = z_i^k$ для всех i и k решение задач (5)-(8) и (9)-(13) позволяет определить решение задачи (1)-(4).

В случае выполнения неравенства (12) на каждой итерации алгоритма

сумма ($t_v + \sum_{k=1}^{N} \widetilde{t}_v^k$) будет стремиться к нулю при $\alpha_v, \widetilde{\alpha}_v^k \to \infty$ и $v \to \infty$

Действительно, при выполнении неравенства (15) на всех итерациях алгоритма получаем:

$$t_{v+1} + \sum_{k=1}^{N} \widetilde{t}_{v+1}^k \le \varepsilon \left(t_v + \sum_{k=1}^{N} \widetilde{t}_v^k\right) \le \varepsilon^v \left(t_1 + \sum_{k=1}^{N} \widetilde{t}_1^k\right),$$

что приводит к выполнению условия $(t_{v+1} + \sum_{k=1}^{N} \widetilde{t}_{v+1}^k) \to 0$ при $v \to \infty$, когда $\varepsilon < 1$ и $t_1 + \sum_{k=1}^{N} \widetilde{t}_1^k < \infty$. Это в свою очередь приводит к равенствам $y_i^k = z_i^k$ для всех i и k, т. е. к решению исходной задачи.

Если на первой итерации метода хотя бы одна из задач (9)-(11) или задача (5)-(6) не имеет решения при $\alpha_0 = 0$ и $\widetilde{\alpha}_0^k = 0$, то и задача (1)-(4) также не

имеет решения.

Действительно, допустимое множество задачи (5)-(6) ограничено, поскольку в этой задаче имеются ограничения $0 \le y_i^k \le \widetilde{M}_i^k$, и если задача (5)-(6) не имеет решения при $\alpha_0 = 0$, то ограничения (6) должны быть несовместными. Это возможно, когда несовместными являются и ограничения (2) задачи (1)-(4), которые аналогичны ограничениям (6). Если же хотя бы одна из задач (9)-(11) при каком-либо k и $\widetilde{\alpha}_0^k = 0$ не имеет решения, то ограничения (10)-(11) должны быть несовместными, что возможно, когда несовместными являются и ограничений (3)-(4) задачи (1)-(4), которые аналогичны ограничениям (10)-(11).

Если на первой итерации метода задача (5)-(6) и задачи (9)-(11) будут иметь решения при $\alpha_0 = 0$ и $\widetilde{\alpha}_0^k = 0$, то и на всех последующих итерациях метода задачи (5)-(8) и (9)-(13) должны иметь решения.

Действительно, если задача (5)-(6) имеет решения при $\alpha_0 = 0$, то включение в эту задачу ограничений (7)-(8) не может привести к несовместности ограничений (6). Это справедливо, поскольку при любых $0 \le z_i^k \le \widetilde{M}_i^k$ всегда найдутся такие $t_v \ge 0$, $u_i^k \ge 0$, $w_i^k \ge 0$, что ограничения (6)-(8) будут выполняться для всех значений y_i^k, для которых выполняются и ограничения (6). Такие же рассуждения справедливы и для задач (9)-(11) и (9)-(13).
Если на какой-либо итерации метода после увеличения штрафных коэффициентов α_v и $\widetilde{\alpha}_v^k$ не выполняется условие (12), то согласно вычислительной схеме метода в модель (5)-(8) включается дополнительное ограничение $t_v \le \varepsilon\, t_{v-1}$, а в каждую k-ю модель (9)-(13) – ограничение $\widetilde{t}_v^k \le \varepsilon\, \widetilde{t}_{v-1}^k$.

Если задача (5)-(8) и все задачи (9)-(11) с такими дополнительными ограничениями окажутся несовместными, то и задача (1)-(4) будет несовместной. Если хотя бы одна из этих задач окажется совместной, то неравенство (12) будет выполняться на этой итерации метода.

В тех же случаях, когда задача (5)-(6) и задачи (9)-(11) на первой итерации метода при $\alpha_0 = 0$ и $\widetilde{\alpha}_0^k = 0$ имеют решения, а модели (5)-(8) и (9)-(13) при введении дополнительных ограничений $t_{v+1} \le \varepsilon\, t_v$, $\widetilde{t}_{v+1}^k \le \varepsilon\, \widetilde{t}_v^k$ на всех итерациях, где эти ограничения вводятся, оказываются совместными, то неравенства (15) будут выполняться на всех итерациях метода. Это приводит к выполнению условия $(t_{v+1} + \sum_{k=1}^{N} \widetilde{t}_{v+1}^k) \to 0$ при $v \to \infty$, когда в соответствии со схемой метода $\varepsilon < 1$, а $t_1 + \sum_{k=1}^{N} \widetilde{t}_1^k < \infty$. Выполнение условия

$(t_{v+1}+\sum_{k=1}^{N}\widetilde{t}_{v+1}^{\,k})\to 0$ в свою очередь приводит к решению исходной задачи.

Теорема доказана.

5. Согласование работы отдельных программных модулей

Организация согласованного функционирования уже разработанных программных модулей, которые использовались для моделирования и управления в различных подсистемах некоторой системы, может производиться в соответствии со следующей схемой.

Сначала выделяются общие ресурсы, которые используются всеми подсистемами исследуемой системы и на которые накладываются общие ограничения.
Затем в моделях, описывающих функционирование отдельных подсистем, выделяются переменные, с использованием которых в каждой конкретной подсистеме учитывается использование общих ресурсов. Для таких переменных, которые по аналогии с переменными моделей (9)-(13) условно можно обозначить через z_i^k, вводятся параметры y_i^k. Кроме того, для таких переменных в модели, описывающей подсистемы исследуемой системы, вводятся ограничения вида (11) и (13) с вспомогательными переменными u_i^k, w_i^k, u_i^k, w_i^k, t_v и $\widetilde{t}_v$, а оптимизируемый функционал приводится к виду (9).

Для тех подсистем исследуемой системы, для которых работающих моделей и их программной реализации еще не было разработано, целесообразно строить модели, имеющие структуру аналогичную моделям (9)-(13).

Затем для организации согласованного моделирования системы строится координирующая модель по аналогии с моделью (5)-(8). Это позволяет использовать описанный выше алгоритм для определения оптимальных параметров функционирования системы с использованием уже имеющегося программного обеспечения, которое использовалось при моделировании отдельных подсистем системы.

Здесь также следует отметить, что данный метод может быть использован и для решения задач, у которых функционал имеет вид:

$$J = \min\{\sum_{k=1}^{N}\left(\varphi^k(x^k, y^k) + \sum_{i=1}^{n_k} c_i^k x_i^k\right) + \sum_{k=1}^{N}\sum_{i=1}^{m_k} \hat{c}_i^k y_i^k\},$$

где $\varphi^k(x^k, y^k)$ - вогнутые функции по переменным $x_1^k, \ldots, x_{n_k}^k, y_1^k, \ldots, y_{m_k}^k$, а ограничения на переменные имеют такой же вид, как в задаче (1)-(4). При решении данной задачи с использованием описанного метода координирующую задачу целесообразно строить таким образом, чтобы ее функционал имел следующий вид:

$$\tilde{J} = \min\{\alpha_v t_v\},$$

а функционалы отдельных задач были бы представлены в виде:

$$\hat{J}_k = \min\{\left(\varphi^k(x^k, y^k) + \sum_{i=1}^{n_k} c_i^k x_i^k\right) + \sum_{k=1}^{N}\sum_{i=1}^{m_k} \hat{c}_i^k y_i^k\}.$$

Это позволяет иметь в качестве координирующей задачи задачу линейного программирования, что значительно облегчает получение решения, особенно в тех случаях, когда часть отдельных задач также является задачами линейного программирования.

Литература

1. *Рыжиков Ю.И.* Имитационное моделирование. Теория и технологии. СПб.: КОРОНА принт, 2004.

2. Прицкер А. Введение в имитационное моделирование и язык СЛАМ II. М.: Мир, 1987.

3. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Высшая школа, 1998.

4. Калашников В.В. Организация моделирования сложных систем. М.: Знание, 1982.

5. Цвиркун А.Д., Акинфиев В.К., Филиппов В.А. Имитационное моделирование в задачах синтеза структуры сложных систем. М.: Наука, 1985.

6. Поляк Б.Т. Введение в оптимизацию. М.: Наука, 1983.

7. Мину М. Математическое программирование. Теория и алгоритмы. М.: Наука, 1990.

8. Базара М., Шетти К. Нелинейное программирование. Теория и алгоритмы. М.: Мир, 1982.

9. Умнов А.Е. Распределенное моделирование объектов сложной структуры. М.: Международный НИИ проблем управления, 1980, 45 с.

10. Мелкишев В.Н., Хоботов Е.Н. Об одном подходе к построению систем распределенного моделирования. Труды XLVI научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук». Часть VII. Москва, ноябрь 2003 г., 108-111.

11. Хоботов Е.Н. Моделирование в задачах реинжиниринга производственных систем // АиТ. 2001. № 8.