

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»
Институт новых материалов и технологий
Кафедра «Теплофизика и информатика в металлургии»

ДОПУСТИТЬ К ЗАЩИТЕ:

Бякову Марию Андреевну

Зав. кафедрой Н.А. Спирин

«___» _____ 2018 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Разработка информационной системы расчета оптимального распределения топливно-энергетических ресурсов в группе доменных печей

МД 09.04.02 000 000 601 ПЗ

Руководитель

профессор, д.т.н.
(должность, уч. ст., зван.)

(подпись)

В.В. Лавров

(расшифровка подписи)

Нормоконтролер

доцент, к.т.н.
(должность, уч. ст., зван.)

(подпись)

Е.В. Киселев

(расшифровка подписи)

Студент

НМТМ-263901
(группа.)

(подпись)

М.А. Бякова

(расшифровка подписи)

Екатеринбург,
2018 г.

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт Новых материалов и технологий
Кафедра Теплофизика и информатика в металлургии
Направление (магистратура) 09.04.02 – Информационные системы и технологии

УТВЕРЖДАЮ:

Зав. кафедрой _____ (Н.А. Спирин)
(подпись)

« _____ » _____ 2018 г.

ЗАДАНИЕ
на выполнение магистерской диссертации

студентке Бяковой Марии Андреевне группы НМТМ-263901
(фамилия, имя, отчество)

1 Тема магистерской диссертации Разработка информационной системы расчета оптимального распределения топливно-энергетических ресурсов в группе доменных печей

Утверждена распоряжением по ИНМТ от « _____ » _____ 2018 г. № _____

2 Руководитель Лавров В.В., профессор, доцент, д.т.н.
(Ф.И.О., должность, ученое звание, ученая степень)

3 Исходные данные к работе _____

3.1) структура комплекса АРМ «Технолог доменного цеха»;

3.2) отчетные показатели доменного производства ПАО «ММК»;

3.3) требования пользователей к системе;

3.4) результаты преддипломной практики;

3.5) информационные ресурсы сети Интернет

4 Содержание магистерской диссертации

4.1) состояние вопроса, постановка цели и задач ВКР;

4.2) проектирование системы (выбор среды для создания системы тестирования, алгоритмическое и функциональное проектирование, создание архитектуры системы тестирования);

4.3) реализация системы тестирования

5 Перечень демонстрационных материалов _____

Постановка задачи, архитектура информационной системы, структура функциональных блоков программного обеспечения, блок-схема работы пользователя с системой, фрагменты пользовательского интерфейса, примеры реализации системы.

6 Календарный план

Наименование этапов выполнения работы	Срок выполнения этапов работы	Отметка о выполнении
<i>Разработка проектной части</i>	01.04 – 15.04.2018 г.	
<i>Оформление пояснительной записки</i>	12.05 – 23.05.2018 г.	
<i>Создание демонстрационных материалов</i>	26.05 – 29.05.2018 г.	
<i>Оформление магистерской диссертации</i>	до 29.05.2018 г.	

Руководитель _____ (В.В. Лавров)
(подпись)

Задание приняла к исполнению _____ (М.А. Бякова)
(подпись)

7 Выпускная работа закончена « ____ » _____ 2018 г.

Пояснительная записка и все материалы просмотрены.

Считаю возможным допустить Бякову Марию Андреевну
к защите ее магистерской диссертации в Государственной экзаменационной комиссии.

Руководитель _____ (В.В. Лавров)
(подпись)

8 Допустить Бякову Марию Андреевну к защите магистерской диссертации в Государственной экзаменационной комиссии (протокол заседания кафедры № _____ от « ____ » _____ 2018 г.)

Зав. кафедрой _____ (Н.А. Спирин)
(подпись)

РЕФЕРАТ

Пояснительная записка содержит 70 с., 24 рис., 10 табл., 30 библиогр. наименований.

ДОМЕННОЕ ПРОИЗВОДСТВО, ПРИРОДНЫЙ ГАЗ, РАСПРЕДЕЛЕНИЕ ЭНЕРГОРЕСУРСОВ, ОПТИМИЗАЦИЯ, МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ПРОГРАММНЫЙ ПРОДУКТ.

Диссертационная работа посвящена разработке программного обеспечения системы оптимального распределения топливно-энергетических ресурсов в группе доменных печей.

В ходе работы рассмотрены основные этапы разработки программного обеспечения: анализ предметной области; создание архитектуры программного обеспечения, пользовательского интерфейса, структуры базы данных; разработка алгоритмического обеспечения и справочной документации; подготовка дистрибутива.

Результатом работы является программный продукт, предназначенный для инженерно-технологического персонала доменного цеха металлургического комбината. Результаты работы могут быть использованы также в учебном процессе для обучения бакалавров и магистрантов по направлениям «Металлургия» и «Информационные системы и технологии».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 СОСТОЯНИЕ ВОПРОСА. ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ МАГИСТЕРСКОЙ ДИССЕРТАЦИИ.....	6
1.1 Структура информационных систем металлургического предприятия для решения технологических задач доменного производства	6
1.2 Математическая модель оптимального распределения топливно- энергетических ресурсов в группе доменных печей	2
1.3 Требования к программному обеспечению.....	7
1.4 Программные средства решения оптимизационных задач	10
1.5 Постановка цели и задач магистерской диссертации	11
2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКИХ РЕСУРСОВ В ГРУППЕ ДОМЕННЫХ ПЕЧЕЙ.....	13
2.1 Разработка функциональной модели программного обеспечения	13
2.2 Разработка архитектуры подсистемы	15
2.3 Реализация модуля решения задачи оптимизации	17
2.4 Разработка базы данных	21
2.5 Создание программного обеспечения в системе управления версиями Atlassian Bitbucket.....	8
2.6 Разработка пользовательского интерфейса	9
2.7 Вывод по главе	13
3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	14
3.1 Установка и настройка программного продукта	14
3.2 Функциональные возможности программного продукта.....	14
3.3 Описание работы программного обеспечения.....	14
3.4 Вывод по главе	19
ЗАКЛЮЧЕНИЕ	20
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	21
ПРИЛОЖЕНИЕ А	25

ВВЕДЕНИЕ

Черная металлургия является одним из крупнейших промышленных секторов в современной экономике. Производство черных металлов обладает исключительной значимостью – ее продукция используется повсеместно, многие отрасли промышленности напрямую зависят от работы металлургических предприятий.

Особое место в комплексе технологических операций получения металлопродукции отводится доменному переделу как самому энергоёмкому и сложному, на долю которого приходится до 50 % топлива, используемого в чёрной металлургии. В 2015 году мировое производство чугуна составило 1 622 млн. т. При этом было использовано 600 млн. т металлургического кокса, который был получен путем переработки свыше 820 млн. т дефицитных коксующихся углей [1]. Поскольку себестоимость каждой тонны чугуна на 48–50% определяется стоимостью кокса, то вся история доменного процесса связана с поиском путей экономии дорогостоящего кокса. Один из возможных эффективных способов снижения удельного расхода кокса – использование комбинированного дутья, обогащенного природным газом и кислородом.

Цены на кокс и природный газ, согласно экономическим прогнозам, продолжают расти в ближайшей и долгосрочной перспективе. При заданном на доменный цех общем расходе этих ресурсов целесообразно иметь оперативную методику оценки эффективности использования топливно-энергетических ресурсов на отдельных доменных печах и осуществлять их оптимальное распределение. Это особенно актуально для крупных доменных цехов, включающих до 10 доменных печей, выплавляющих в сутки до 30 тыс. тонн чугуна, для чего используется около 45 тыс. тонн железорудного сырья, 13 тыс. тонн кокса, 3,5 млн. м³ технологического кислорода, до 4,0 млн. м³ природного газа [2].

Эффективность применения комбинированного дутья определяется факторами теплоэнергетического, технологического, экономического характера

и внешними условиями. Наличие многих факторов и критериев, определяющих эффективность использования комбинированного дутья, а также ограничений на расходы топливно-энергетических ресурсов существенно усложняет задачу по определению оптимальных параметров дутья, при которых достигаются наилучшие технико-экономические показатели работы как отдельных доменных печей, так и группы печей или цеха в целом. Решить эту задачу можно только на основе широкого использования современных методов математического моделирования и разработок на этой основе автоматизированных систем оптимизации распределения энергоресурсов [3–5]. Анализ состояния вопроса по реально используемым математическим моделям в практике ведения технологии доменной плавки показывает: в настоящее время разрыв между потенциальными возможностями средств автоматизации и реальными возможностями используемого программного обеспечения огромен [2, 6, 7].

Автоматизированная подсистема оптимального распределения инжестируемого топлива в режиме «советчик» позволит рекомендовать для инженерно-технологического персонала оптимальные параметры комбинированного дутья для каждой печи при изменении индивидуальных режимных параметров их работы, объема имеющихся топливно-энергетических ресурсов и конъюнктуры рынка [2, 6, 7].

1 СОСТОЯНИЕ ВОПРОСА. ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

1.1 Структура информационных систем металлургического предприятия для решения технологических задач доменного производства

Архитектура автоматизированных систем, как правило, иерархическая с несколькими уровнями управления. При ее создании была использована концепция четырехуровневого построения автоматизированной системы, которая в самых общих чертах сохраняется и в настоящее время.

- Первый (нижний) уровень управления – датчики, преобразователи, аппаратура ручного управления.

- Второй уровень (уровень локального управления), включающий ручное дистанционное управления, а также дублирующее цифровое.

- Третий уровень (уровень цифрового локального управления), включающий совокупность микроЭВМ и программно-логических регуляторов, позволяющий действовать в полуавтоматическом режиме, где оператор только дополнительно изменяет задания.

- Четвертый (верхний) уровень включает сбор, контроль и хранение информации с помощью управляющей вычислительной машины (УВМ).

На верхнем уровне ЭВМ используется для решения планово-производительных и экономических задач коксохимического, агломерационного и доменного цехов. Средний – две миниЭВМ, выполняющие общие задачи управления технологическим процессом: обработка информации, протоколирование, статистический анализ, диагностика и оптимизация процесса. На нижнем уровне используются микроЭВМ (микропроцессоры) и программируемые контроллеры для сбора и первичной обработки информации, для прямого управления локальным участком или процессом.

Рациональный объем автоматизации доменных печей включает также три взаимосвязанных функциональных уровня:

1. Нижний уровень – средства автоматического контроля и диагностики основных входных, технологических и выходных параметров доменной плавки.

2. Средний уровень – локальные системы автоматического регулирования, стабилизации и диагностики основных технологических параметров.

3. Верхний уровень – распределенные автоматизированные подсистемы управления шихтовкой и дутьем, оценки и управления доменным процессом.

Архитектура автоматизированной информационной системой технологического обслуживания и управления доменного цеха включает следующие основные подсистемы (рисунок 1.1):

- распределенная подсистема контроля, сигнализации и локального управления (РСКУ);
- агрегатная подсистема;
- диспетчерская подсистема технологического персонала доменной печи;
- диспетчерская подсистема доменного цеха;
- инженерная подсистема;
- подсистема координации и управления информационной системой.

1.1.1 Распределённая подсистема контроля и локального управления. Входная информация о работе каждой доменной печи разделена на блоки.

1. Параметры, требующие периодичности опроса 2- 5 секунд (критические параметры), такие как анализ схода шихтовых материалов, время срабатывания подач, перепады давления по высоте печи, давление колошниковога газа, горячего дутья, работа ВРШ, дискретные сигналы и т.д.

2. Блок остальных входных данных, автоматически контролируемых непосредственно на печи: анализ состава колошниковога газа, температур периферийных термопар, колошниковога газа по газоотводам, влажность, температура, расход и состав горячего дутья, расходы природного газа и теологического кислорода, расходы кокса и железорудных материалов и т.п. Периодичность опроса состояния этих автоматически контролируемых параметров составляет 30-60 секунд.

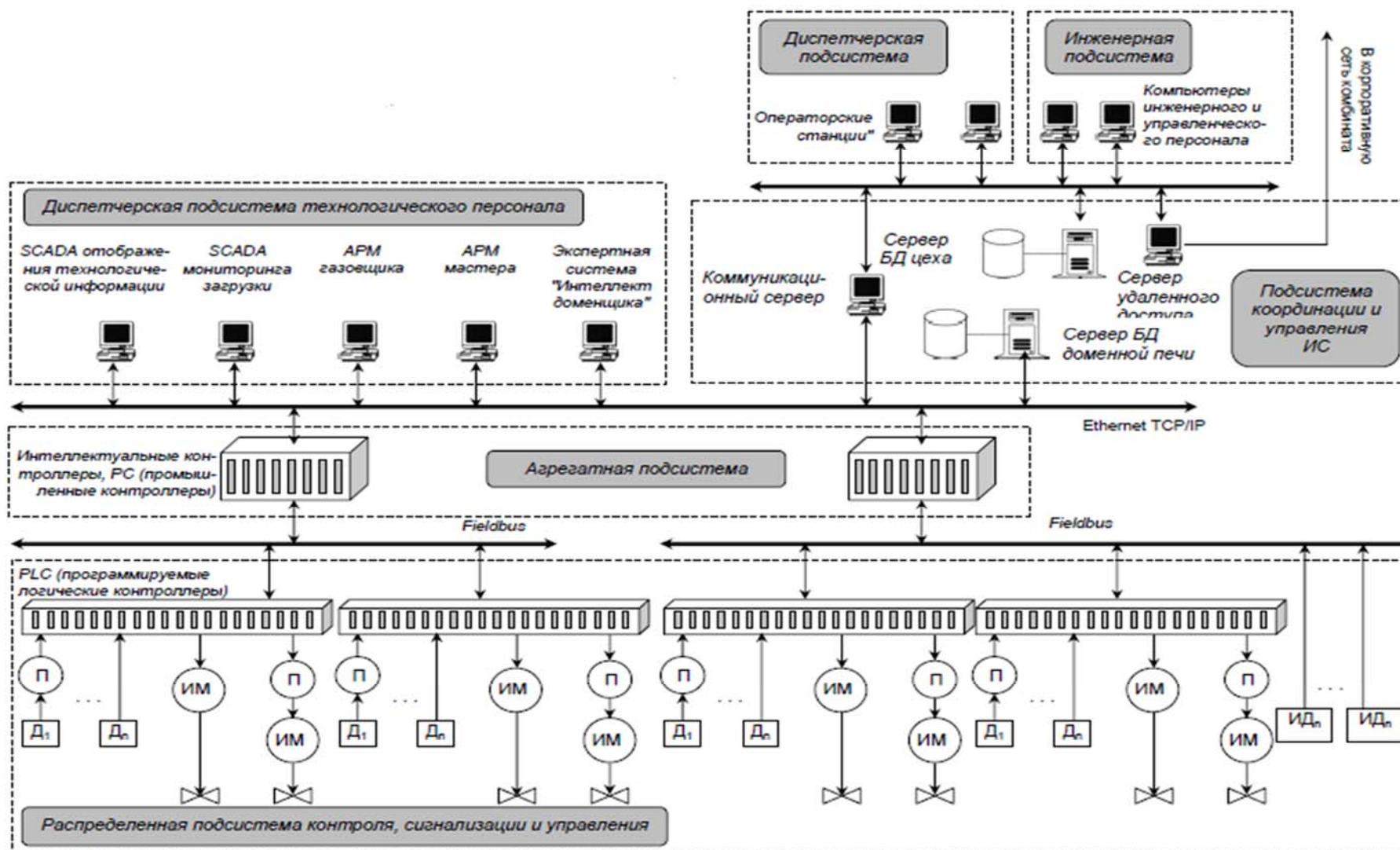


Рисунок 1.1 – Структура автоматизированной информационной системы технического обслуживания и управления доменного цеха

3. Автоматический или полуавтоматический ввод контролируемых данных в ПК рабочего место газовщика: химический состав железорудного сырья, его механическая прочность, содержание мелочи; технический анализ кока, его прочность; состав жидких продуктов плавки и т.п.

Структура Автоматизированной информационной системы доменной печи №1 ММК представлена на рисунке 1.2 [7].

1.2 Математическая модель оптимального распределения топливно-энергетических ресурсов в группе доменных печей

1.2.1 Постановка задачи. В состав доменного цеха входит N печей, на каждой из которых осуществляется инжектирование природного газа (ПГ) с расходами $V_i^{\text{ПГ}}$ ($i = 1, \dots, N$), $\text{м}^3/\text{ч}$. Эффективность инжектируемого топлива оценивается эквивалентом замены кокса e_i ($\text{кг}/\text{м}^3$), физический смысл которого – это количество кокса, сэкономленного на i -й печи при вдувании 1 м^3 природного газа. Задан лимит на подачу природного газа в цех $V_{\Sigma}^{\text{ПГ}}$ ($\text{м}^3/\text{ч}$), а также прямые ограничения по расходу газа на каждую доменную печь, в частности, $V_{i,\min}^{\text{ПГ}} \leq V_i^{\text{ПГ}} \leq V_{i,\max}^{\text{ПГ}}$. Установлены также цеховой ресурс по расходу кокса и требуемая производительность доменного цеха по чугуна, которые равны соответственно K_{Σ} (т кокса/ч) и Π_{Σ} (т чугуна/ч).

В качестве технологических ограничений на каждую из доменных печей приняты интервалы: по теоретической температуре горения на фурмах, по содержанию кремния в чугуне, по отношению теплоемкостей потоков шихты и газа, по содержанию серы в чугуне. Требуется распределить подачу природного газа между печами доменного цеха таким образом, чтобы при существующих ограничениях была достигнута максимальная эффективность его использования в целом по цеху.

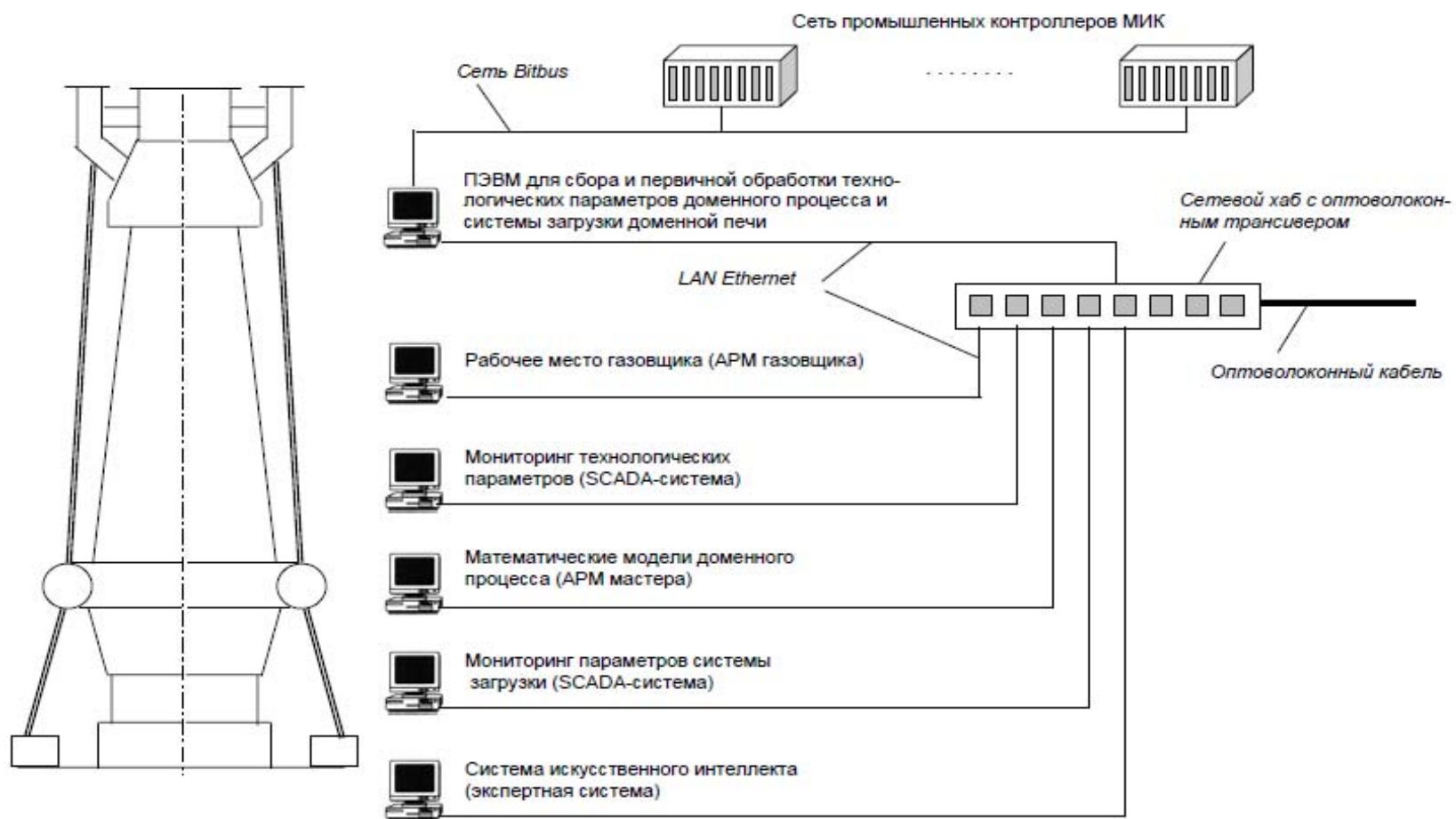


Рисунок 1.2 – Структура автоматизированной информационной системы доменной печи №1 ММК

Постановка задачи оптимального распределения природного газа в группе доменных печей проиллюстрирована на рисунке 1.3 [2, 6, 7]. Последовательность решения задачи оптимизации распределения природного газа отражена в схеме рисунок 1.4 [6], где очевидны назначение и функции отдельных этапов.

Особенность решения сложных задач математического программирования связана с тем, что в процессе решения могут возникнуть такие случаи, когда ограничения, наложенные на работу цеха в целом, или технологические ограничения на работу отдельных печей противоречивы, т.е. отсутствует область допустимых решений. В связи с этим в алгоритме решения предусмотрен этап анализа решения задачи. В случае отсутствия области решения задачи и противоречивости условий приходится воспользоваться этапом коррекции, т.е. оценить корректность ограничений, надёжность исходной информации и т.д.

1.2.2 Математическая модель. При математической постановке и решении задач оптимизации распределения топливно-энергетических ресурсов требуется не только изучение статических характеристик, описывающих влияние изменений условий доменной плавки на технико-экономические показатели работы печей, но и выполнение детального анализа для математического описания как внешних, так и внутренних ограничений [2, 6, 7]. Кроме того, на практике приходится оценивать техническое состояние печи и оборудования, объем расходов природного газа, качество сырья и другие факторы, которые учитываются только методом экспертных оценок. Поскольку общее количество топливно-энергетических ресурсов (мощность источников) задано, то эти рекомендации должны учитывать и общий лимит ресурсов, выделяемый цеху.

Используемый подход базируется на следующих положениях: в случае чрезвычайной сложности исходной задачи, необходимости учёта большого числа нелинейных ограничений типа неравенств, при относительно небольших колебаниях параметров относительно базовых значений целесообразно использовать принцип малых отклонений и свести задачу оптимизации к линейному математическому программированию [2; 5; 7].

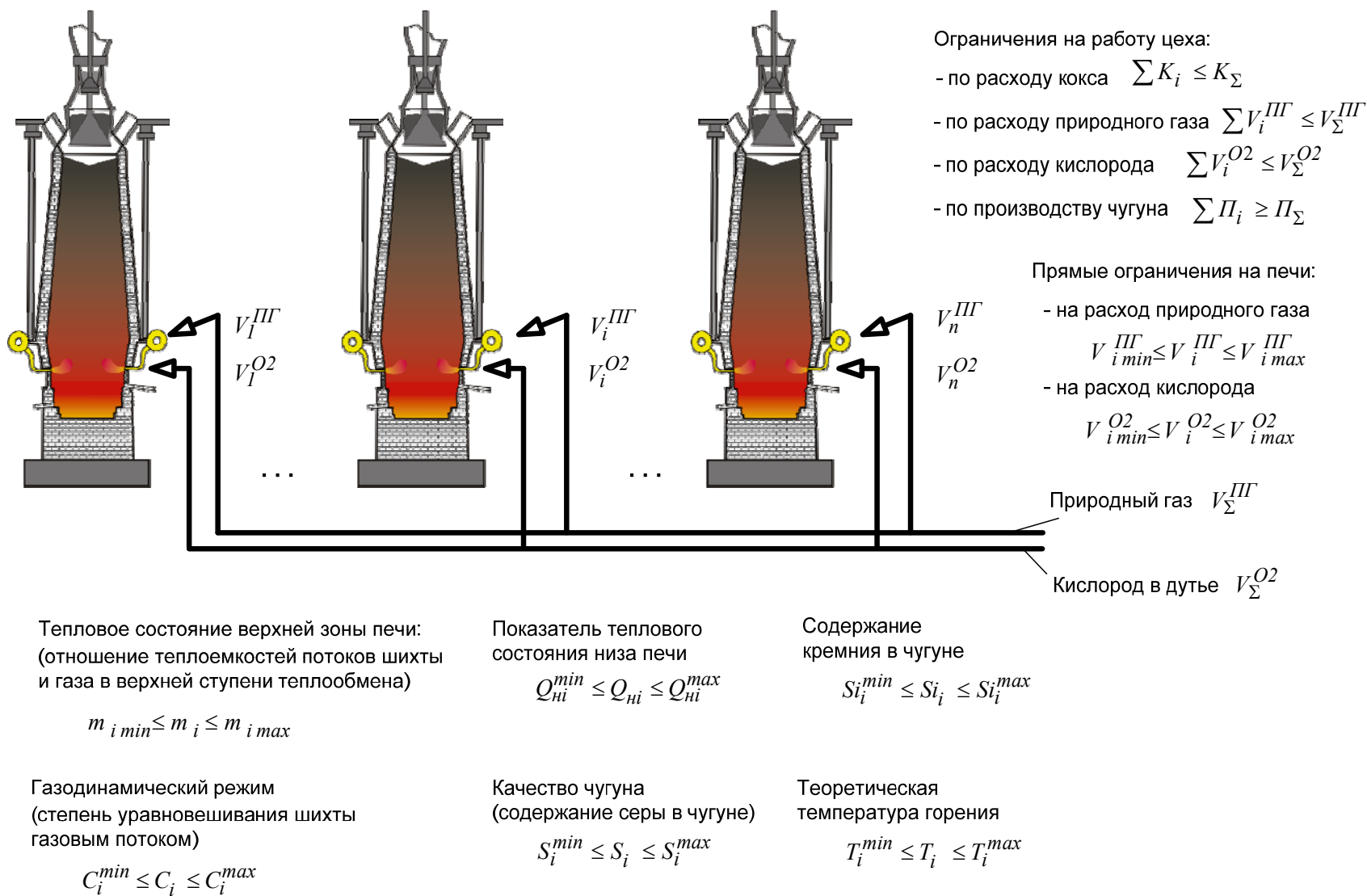


Рисунок 1.3 – Постановка задачи распределения природного газа в группе доменных печей

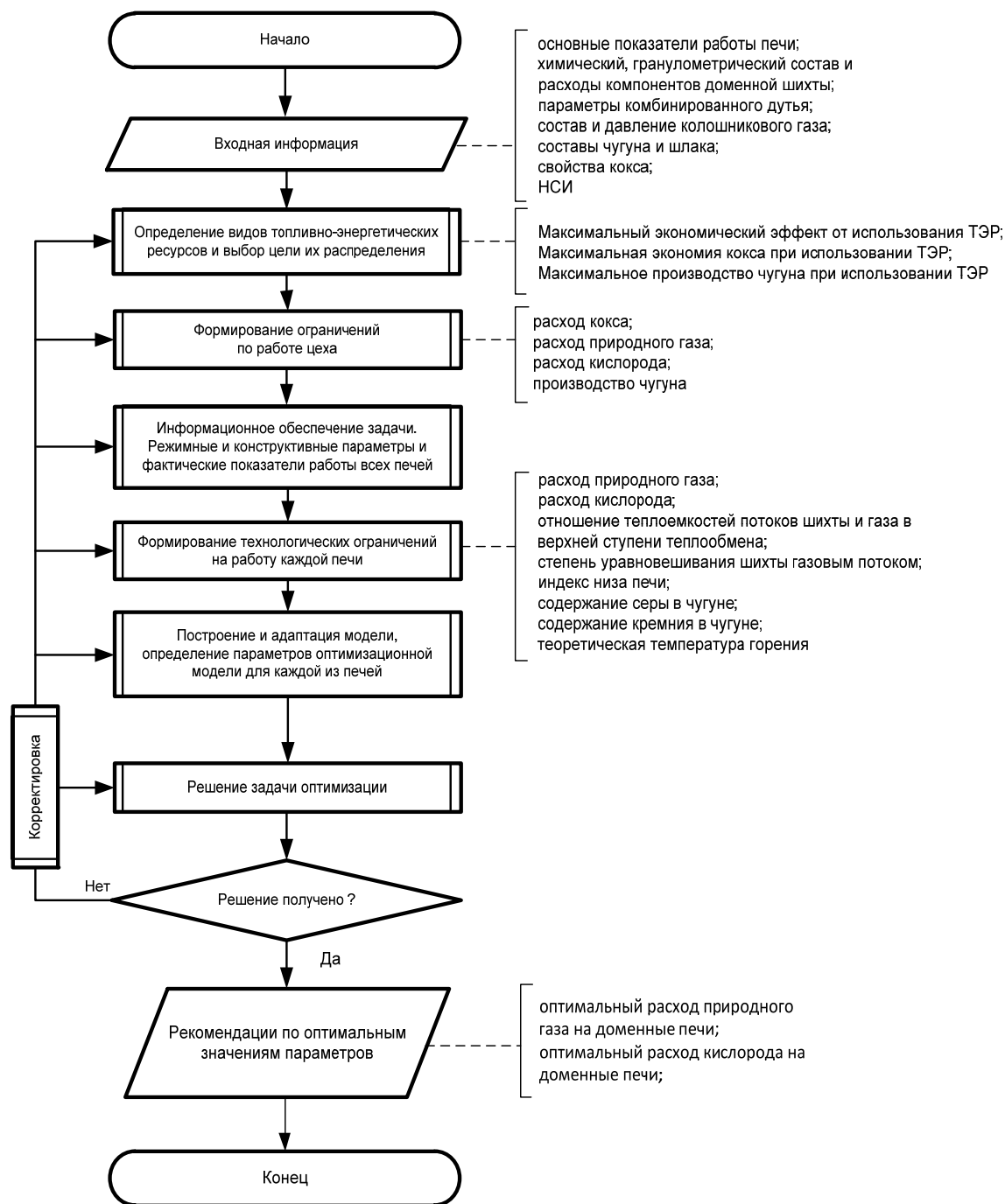


Рисунок 1.4 – Последовательность решения задачи оптимизации распределения топливно-энергетических ресурсов в группе доменных печей

$$Z = \sum_{i=1}^N \left\{ 0,5(e_i C_K - C_{ПГ}) + 0,5 \cdot C_{ПГ} (\Delta \Pi_i^{ПГ} - e_i \Delta \Pi_i^K) \right\} \cdot V_i^{ПГ} \rightarrow \max, \quad (1.1)$$

где N – число печей в рассматриваемой группе (цехе);

e_i – эквивалент замены кокса, кг кокса/(м³ ПГ);

C_K – стоимость кокса, руб/кг;

$C_{ПГ}$ – стоимость природного газа, руб/м³;

$C_{П}$ – условно постоянный коэффициент, руб/т чугуна (принять $C_{П}=1$);

$\Delta\Pi_i^{ПГ}$ – прирост производства чугуна при изменении расхода природного газа на i -й печи, т чугуна/(м³ ПГ);

$\Delta\Pi_i^K$ – прирост производства чугуна при изменении расхода кокса на i -й печи, т чугуна/(кг кокса);

$V_i^{ПГ}$ – расход природного газа на i -й доменной печи, м³/ч.

1.2.2.1 Ограничения в целом по цеху

По расходу природного газа в целом по цеху

$$\sum_{i=1}^N V_i^{ПГ} \leq V_{\Sigma}^{ПГ}, \quad (1.2)$$

где $V_{\Sigma}^{ПГ}$ – резерв по расходу природного газа в целом по цеху, м³/ч.

По расходу кокса в целом по цеху

$$\sum_{i=1}^N \{K_{0i} + 0,001(V_{i0}^{ПГ} - V_i^{ПГ})e_i\} \leq K_{\Sigma}, \quad (1.3)$$

где K_{0i} – расход кокса на i -ю печь в базовом периоде, т/ч;

K_{Σ} – запасы кокса по цеху, т/ч;

$V_{i0}^{ПГ}$ – расход природного газа на i -й печи в базовом периоде, м³/ч.

По объему производства чугуна в целом по цеху

$$\sum_{i=1}^N \{(V_i^{ПГ} - V_{i0}^{ПГ})\Delta\Pi_i^{ПГ} - e_i(V_i^{ПГ} - V_{i0}^{ПГ})\Delta\Pi_i^K + \Pi_{i0}\} \geq \Pi_{\Sigma}, \quad (1.4)$$

где Π_{Σ} – требуемое производство чугуна в цехе, т/ч;

Π_{i0} – производительность i -й печи по чугуну, т/ч.

1.2.2.2 Технологические ограничения на каждую из печей. В качестве параметров, характеризующих тепловое состояние печи, приняты:

1) Теоретическая температура горения на фурмах:

$$T_i^{\min} \leq (V_i^{ПГ} - V_{i0}^{ПГ})\Delta T_i^{ПГ} + T_i^0 \leq T_i^{\max}, \quad (1.5)$$

где $T_i^{\min}$, $T_i^{\max}$ – минимально и максимально допустимые значения теоретических температур горения, °С;

ΔT_i^{III} – изменение теоретической температуры горения при увеличении расхода природного газа на печи на м³/ч, т.е. °C/ (м³/ч).

2) *Содержание кремния в чугунах на каждую из печей:*

$$Si_i^{\min} \leq (V_i^{III} - V_{i0}^{III}) \Delta Si_i^{III} - e_i (V_i^{III} - V_{i0}^{III}) \Delta Si_i^K + [\Delta \Pi_i^{III} (V_i^{III} - V_{i0}^{III}) - e_i (V_i^{III} - V_{i0}^{III}) \Delta \Pi_i^K] \Delta Si_i^II + Si_{i0} \leq Si_i^{\max} \quad (1.6)$$

где $Si_i^{\min}$, $Si_i^{\max}$ – минимально и максимально допустимые значения содержания кремния в чугунах, %;

ΔSi_i^{III} – приращение содержания кремния в чугунах (в процентах) при увеличении расхода природного газа м³

ΔSi_i^K – приращение содержания кремния в чугунах (в процентах) при увеличении расхода кокса на кг/ч;

ΔSi_i^II – приращение содержания кремния в чугунах (в процентах) при увеличении производительности печи на т/ч;

Si_{i0} – содержание кремния в чугунах на i -й доменной печи в базовом периоде, %.

3) *Отношение теплоёмкостей потоков шихты и газа:*

$$m_i^{\min} \leq (V_i^{III} - V_{i0}^{III}) \Delta m_i^{III} - e_i (V_i^{III} - V_{i0}^{III}) \Delta m_i^K + [\Delta \Pi_i^{III} (V_i^{III} - V_{i0}^{III}) - e_i (V_i^{III} - V_{i0}^{III}) \Delta \Pi_i^K] \Delta m_i^II + m_{i0} \leq m_i^{\max} \quad (1.7)$$

где $m_i^{\min}$, $m_i^{\max}$ – минимально и максимально допустимые значения содержания кремния в чугунах, %;

Δm_i^{III} – приращение содержания кремния в чугунах (в процентах) при увеличении расхода природного газа м³/ч;

Δm_i^K – приращение содержания кремния в чугунах (в процентах) при увеличении расхода кокса на кг/ч;

Δm_i^II – приращение содержания кремния в чугунах (в процентах) при увеличении производительности печи на т/ч;

m_{i0} – содержание кремния в чугунах на i -й доменной печи в базовом периоде, %.

4) Учет полноты конверсии природного газа и газодинамического режима доменной плавки сводится к следующему:

– прямые ограничения на расход природного газа на каждую из печей цеха:

$$V_i^{\min} \leq V_i^{\Pi\Gamma} \leq V_i^{\max}, \quad (1.8)$$

где $V_i^{\min}$, $V_i^{\max}$ – минимально и максимально допустимые значения содержания кремния в чугунае, %;

– степень уравнивания шихты:

$$C_{yi}^{\min} \leq (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) \Delta C_{ni}^{\Pi\Gamma} - e_i (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) \Delta C_{yi}^K + C_{yi0} \leq C_{yi}^{\max}, \quad (1.9)$$

где $C_{yi}^{\min}$, $C_{yi}^{\max}$ – минимально и максимально допустимые степени уравнивания шихты;

$\Delta C_{yi}^{\Pi\Gamma}$ – изменение степени уравнивания шихты при увеличении расхода природного газа на м³/ч;

ΔC_{yi}^K – изменение степени уравнивания шихты при увеличении расхода кокса на кг/ч;

$C_{yi0} - \Delta C_{yi}^K$ – изменение степени уравнивания шихты в базовом периоде, %;

5) Качество выплавленного чугуна оценивается по содержанию серы:

$$S_i^{\min} \leq (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) \Delta S_i^{\Pi\Gamma} - e_i (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) \Delta S_i^K + [\Delta \Pi_i^{\Pi\Gamma} (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) - e_i (V_i^{\Pi\Gamma} - V_{i0}^{\Pi\Gamma}) \Delta \Pi_i^K] \Delta S_i^{\Pi} + S_{i0} \leq S_i^{\max}, \quad (1.10)$$

где $S_i^{\min}$, $S_i^{\max}$ – минимально и максимально допустимые значения содержания серы в чугунае, %;

$\Delta S_i^{\Pi\Gamma}$ – приращение содержания серы в чугунае (в процентах) при увеличении расхода природного газа м³

ΔS_i^K – приращение содержания серы в чугунае (в процентах) при увеличении расхода кокса на кг/ч;

ΔS_i^{Π} – приращение содержания серы в чугунае (в процентах) при увеличении производительности печи на т/ч.

В данной постановке в значительно большей степени учтены ограничения по тепловой работе «верха» (m) и «низа» (Q_n , T) печи, химическому нагреву (содержание в чугуна Si), качеству чугуна (содержание в чугуна S), газодинамическим характеристикам работы печи (C_y). При этом предельно допустимые значения величин, т.е. численные значения параметров $V_{min,i}^{ПГ}$, $V_{max,i}^{ПГ}$, $V_{min,i}^{O2}$, $V_{max,i}^{O2}$, Q_{ni}^{min} , Q_{ni}^{max} , T_i^{min} , T_i^{max} , Si_i^{min} , Si_i^{max} , m_i^{min} , m_i^{max} , S_i^{min} , S_i^{max} , $C_{y,i}^{min}$, $C_{y,i}^{max}$, определяются методом экспертных оценок, что позволяет осуществлять настройку модели на реальный процесс путём формализации опыта инженерно-технического и технологического персонала. Расчёт коэффициентов, определяющих приращения этих параметров при единичных изменениях расходов природного газа, технологического кислорода, кокса и производительности, осуществляется по модели доменного процесса УрФУ [2, 5, 7].

Разработанная модель позволяет решать задачи оптимального распределения расхода природного газа в следующих технологических ситуациях:

- при сохранении объема ресурсов по расходу природного газа для цеха в целом на постоянном уровне;
- при изменении объёма ресурсов по расходу природного газа для цеха в целом;
- для определения максимального объёма ресурсов комбинированного дутья, которое может использовать комплекс доменных печей;
- при изменении ресурса по объёму кокса, имеющегося в распоряжении цеха;
- при изменении требований задания на объем выплавляемого чугуна комплексом доменных печей;
- при изменении конъюнктуры рынка, т.е. требований обеспечения максимума экономии кокса, обеспечения максимума производства,

экономичности работы комплекса печей, цен на топливно-энергетические ресурсы и т.п.

Определение коэффициентов оптимизационной модели включает в себя решение следующих задач:

- оценка состояния каждой из доменных печей в базовый период в целях адаптации оптимизационной модели к конкретным условиям печи;
- расчёт изменения параметров плавки и характеристик теплового, газодинамического, шлакового режимов при последовательном варьировании расходов кокса, природного газа на каждой из печей цеха (прогноз показателей) при постоянстве остальных входных параметров и отсутствии требований, предъявляемых к химическому составу продуктов плавки, физическим характеристикам шлака и т.п.

В итоге определяются коэффициенты, входящие целевую функцию [выражение (1.1)] и в левые части матрицы ограничений [см. выражения (1.2) – (1.10)].

Таким образом, оптимизация распределения расхода природного газа в ряде случаев сопровождается значительной коррекцией параметров комбинированного дутья. При наличии соответствующего математического и программного обеспечения инженерно-технический персонал сможет периодически оперативно прогнозировать (уточнять) степень оптимальности параметров комбинированного дутья и при необходимости вносить необходимые коррективы.

1.3 Требования к программному обеспечению

1.3.1 Требования к структуре и функционированию

1. В программе должна быть предусмотрена возможность загрузки различных вариантов исходных данных. Эти варианты создаются пользователем в процессе работы с программой и сохраняются в виде отдельных файлов на жестком диске. В главном меню программы необходимо для этого

предусмотреть функции «Создать», «Открыть», «Сохранить» и «Сохранить как...». Для пояснения варианта исходных данных в меню предусмотреть функцию «Свойства», где пользователь может в текстовом виде записать заметки к варианту исходных данных.

2. Интерфейс программы должен предусматривать возможность отображения и корректировки исходных данных каждого варианта расчета. Просмотр и корректировка должны быть обеспечены в удобной для пользователя форме, организованы с использованием традиционных элементов управления (списков, всплывающих подсказок, закладок, флажков переключателей и пр.). На всех формах располагать русские слова, без грамматических ошибок. Каждая пользовательская форма, на которой существует возможность корректировки исходных данных, должна иметь защиту от некорректно вводимых данных. Кроме этого, предусмотреть возможность отказа от сделанных исправлений (кнопка «Отмена»). В случае использования сложных диалоговых форм (несколько закладок), предусмотреть кнопку «Применить» для принудительного ввода сделанных исправлений без закрытия окна формы.

3. Процедура расчета оптимизации должна включать в себя фоновое обращение к динамически подключаемой библиотеке (dll), которую надо подготовить в среде Microsoft Visual Studio (C#). Эта dll должна вызывать алгоритм расчета задачи оптимизации с фактическими параметрами и возвращать результат расчета в интерфейсную часть ПО. Решение задачи оптимизации осуществить с помощью модуля MATLAB. Фактические параметры надо подготовить в клиентской части на основе данных, которые введены пользователем в диалоговом окне формы в течение текущего сеанса работы с программой.

4. Результаты расчета должны отображаться на формах (или одной форме, но тогда на разных страницах) в численном и графическом виде. На диаграммах указывать название, а также использовать легенду.

5. В программе должна быть предусмотрена функция формирования отчета по выполненным расчетам. Отчет должен быть предварительно просмотрен пользователем (функция «Предпросмотр») перед его распечаткой или экспортированием в популярные форматы Word, Excel, HTML и пр. Содержание отчета должно включать:

- название задачи;
- дату и время создания отчета;
- ключевые исходные данные, при которых было получено решение;
- результаты расчета в численном и графическом виде.

6. В пункте меню «Справка» необходимо предусмотреть обращение к файлу справочной помощи. Файл справки должен быть контекстно-зависимым, т.е. вызываться из каждой формы и отображать соответствующий раздел справочной помощи. Кроме того, обязательно вставить пункт «О программе».

1.3.2 Требования к надёжности. Надёжность ПП должна обеспечиваться применяемыми методами проектирования, модульной архитектурой построения, высоким уровнем качества программирования, тестированием программного обеспечения.

1.3.3 Дополнительные требования следующие:

- описание задач по проекту в системе управления проектами и задачами Atlassian JIRA;
- программный код проекта расположен в системе управления версиями программного обеспечения Atlassian Bitbucket;
- программная платформа – операционная система Microsoft Windows;
- среда программирования приложения – Microsoft Visual Studio;
- объектно-ориентированная методология разработки;
- расчетный модуль в виде библиотеки dll, которая подключена к клиентскому модулю;
- создание программных тестов;
- надёжность работы, защита от некорректно вводимых данных;

- дружелюбный пользовательский интерфейс, основанный на стандартных элементах управления ОС Windows (обязательно использовать меню, списки, флажки, переключатели, всплывающие подсказки, контекстно-зависимую справку);

- установка программного средства с помощью инсталлятора;
- сохранение конфигурационных настроек во внешнем xml-файле;
- табличная и графическая интерпретация полученных результатов;
- возможность сохранения результатов во внешнем файле отчета, экспорт отчета в Microsoft Office Word, Microsoft Office Excel, HTML и др. популярные форматы.

1.3.4 Требования к функциям (задачам), выполняемым ПО. Программное обеспечение должно выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей. Набор указанных функций определяется во внешнем описании ПП.

1.3.5 Требуемые компьютерные версии. В состав ПП и сопроводительных материалов в компьютерной форме, необходимых для сдачи работы, входят:

- программная реализация (дистрибутив и исходники программного продукта);
- размещение исходного кода проекта в системе Atlassian Bitbucket;
- наличие выполненных задач по проекту в системе управления проектами и задачами Atlassian JIRA;
- дополнительные материалы в электронном виде – файл Microsoft Visio с архитектурой и эскизами пользовательского интерфейса, help-файл в программе Help&Manual 5 (откомпилированный, а также исходники).

1.4 Программные средства решения оптимизационных задач

В настоящее время для компьютеров разработано большое количество программных средств, позволяющих решать комплекс основных инженерных задач в области моделирования и оптимизации технологических процессов.

На рынке программных средств существуют математические пакеты (например, MATLAB [9]), которые специально ориентированы на решение математических задач. Их основное достоинство заключается в наличии сотен и тысяч встроенных математических функций и десятков вычислительных алгоритмов для выполнения расчетов в научной и инженерной практике. Конечно, изучение возможностей таких специализированных пакетов потребует от пользователя немало времени, к тому же необходимые дополнительные затраты на их приобретение. Однако следует учесть, что на выбор метода и компьютерного средства решения задач, как правило, определяющее влияние оказывают вид математической модели [9].

Отличительной особенностью MATLAB от других математических пакетов является наличие средств, которые позволяют использовать функционал, созданный в MATLAB, при разработке автономных приложений. Такая возможность реализуется с помощью продукта MATLAB Compiler. Данный продукт позволяет собирать модули в виде:

- независимое приложение.
- C/C++ библиотеки (DLL в Windows, разделяемые библиотеки в Linux).
- программные компоненты, такие как классы Java, сборки .NET или надстройки Microsoft Excel для использования в других приложениях.

Приложения и библиотеки, созданные с помощью MATLAB Compiler, используют свободно распространяемое исполняемое ядро, называемое MATLAB Compiler Runtime (MCR). Это позволяет запускать приложения, которые используют MATLAB-функции, без необходимости установки дорогостоящей копии MATLAB.

1.5 Постановка цели и задач магистерской диссертации

Целью магистерской диссертации является разработка программного обеспечения информационно-моделирующей системы оптимального распределения топливно-энергетических ресурсов в группе доменных печей.

Информационная система ориентирована на использование специалистами инженерно-технологической группы доменных цехов металлургических предприятий. Она может быть использована в учебном процессе для обучения бакалавров и магистрантов по направлениям «Металлургия» и «Информационные системы и технологии». Разрабатываемая система с одной стороны призвана повысить производительность труда инженеров-технологов, оказать положительное влияние на поддержание стабильной работы агрегатов доменного цеха и улучшить технико-экономические показатели работы доменных печей, а с другой стороны – повысить качество обучения студентов, расширить и углубить знания в области доменного производства.

Для выполнения работы необходимо реализовать следующие задачи:

- изучение предметной области, проработка физической и математической постановки задачи;
- определение требований к разрабатываемой системе, проектирование архитектуры, выбор подходящей для реализации платформы;
- проектирование (концептуальное, даталогическое) и реализация базы данных для хранения исходных данных системы;
- разработка клиентского приложения системы оптимального распределения топливно-энергетических ресурсов в группе доменных печей.

2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ТОПЛИВНО-ЭНЕРГЕТИЧЕСКИХ РЕСУРСОВ В ГРУППЕ ДОМЕННЫХ ПЕЧЕЙ

2.1 Разработка функциональной модели программного обеспечения

Функциональное моделирование – важный этап в разработке ПО. На данном этапе приходит понимание того, как будет устроена программа, последовательность вызова функций программы [12-14]. Данный этап значительно облегчает процесс написания кода программы.

В основу метода, используемого при проектировании функций информационной системы, положены идеи и нотации методики структурного анализа и проектирования IDEF0. Методология IDEF0 является развитием хорошо известного графического языка описания функциональных систем SADT, предложенного Дугласом Россом [12]. Использование методики IDEF0 позволила смоделировать функциональную структуру ИС, выявить производимые им действия и связи между этими действиями, управляющие воздействия и механизмы выполнения функции, что, в конечном итоге, позволило на ранней стадии проектирования предотвратить возможные ошибки.

Диаграммы являются основными рабочими элементами IDEF0-модели. Диаграммы представляют входные-выходные преобразования и указывают правила и средства этих преобразований. Каждая IDEF0-диаграмма содержит блоки (работы) и дуги (линии со стрелками). Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Функциональные блоки на диаграмме изображаются прямоугольниками. Блок представляет функцию или активную часть системы. Каждая сторона блока имеет определенное назначение. Левая сторона предназначена для входов, верхняя – для управления, правая – для выходов, нижняя – для механизмов и вызовов [13, 14].

Функциональная модель разработана с помощью программы Ramus Educational [11]. На рисунке 2.1 приведен общий вид функциональной модели.

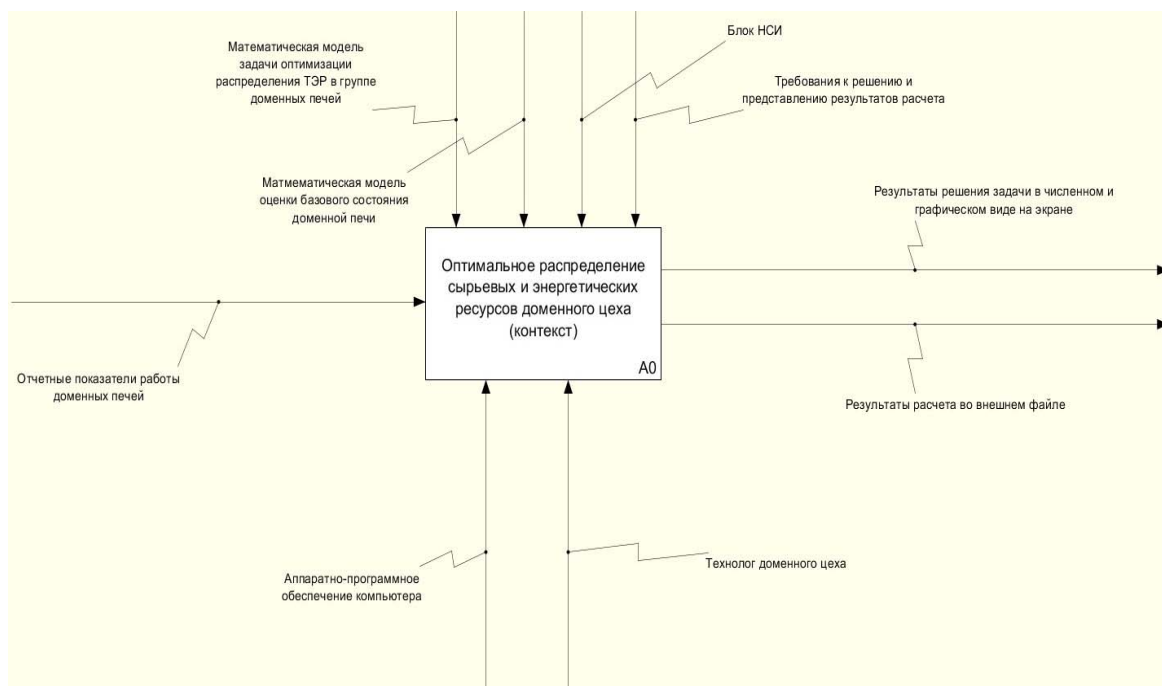


Рисунок 2.1 – Функциональная модель нулевого уровня

На первом уровне функциональной модели выделен один основной поток информации в систему: данные из базы данных.

Интерфейс управления можно разбить на три логических блока: методика расчета, нормативно-справочная информация, а также требования пользователей.

В качестве выходной информации понимается предоставление результатов расчета в табличном и графическом виде для каждой доменной печи.

Декомпозиция является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. При этом уровень детализации процесса определяется непосредственно разработчиком модели.

Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

На рисунке 2.2 представлена взаимосвязь между элементами функциональной модели.

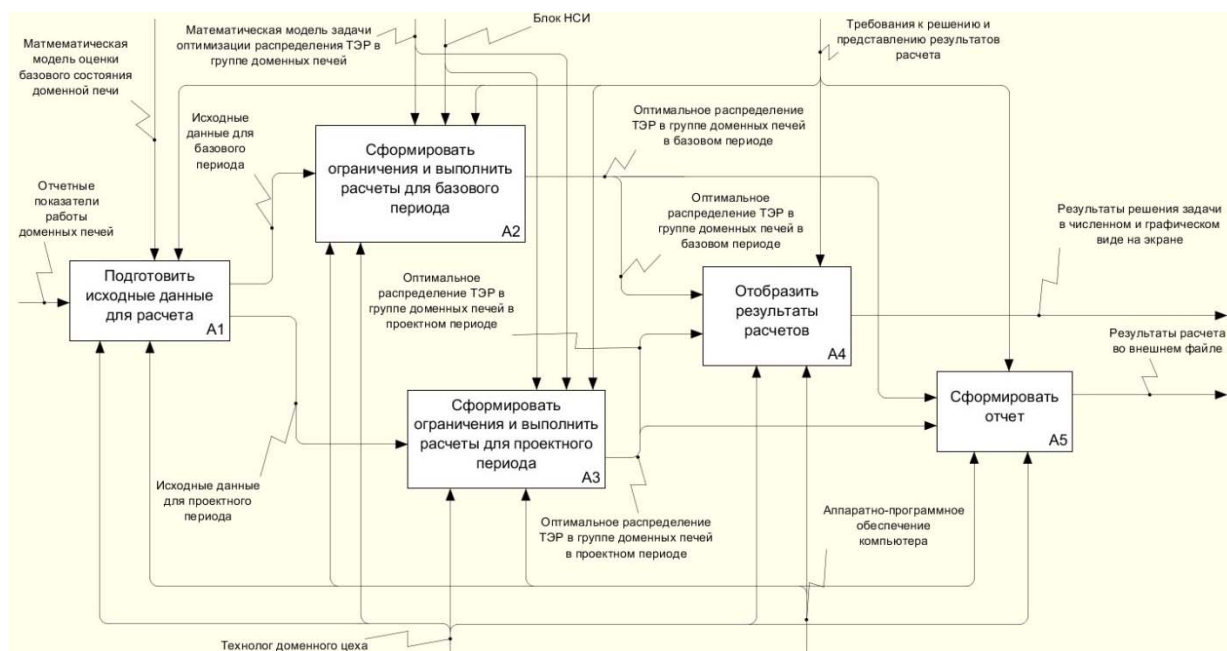


Рисунок 2.2 – Декомпозиция первого уровня функциональной модели

2.2 Разработка архитектуры подсистемы

Для выполнения каждой отдельной функции программы создается свой модуль, который взаимодействует с другими модулями через определенный интерфейс [15-17].

Программное средство должно обеспечивать четыре основные функции:

- загрузка данных для расчета;
- расчет;
- вывод результатов;
- справка.

Архитектура проектируемого программного средства с помощью программы Microsoft Visio 2013 и представлена на рисунке 2.3.

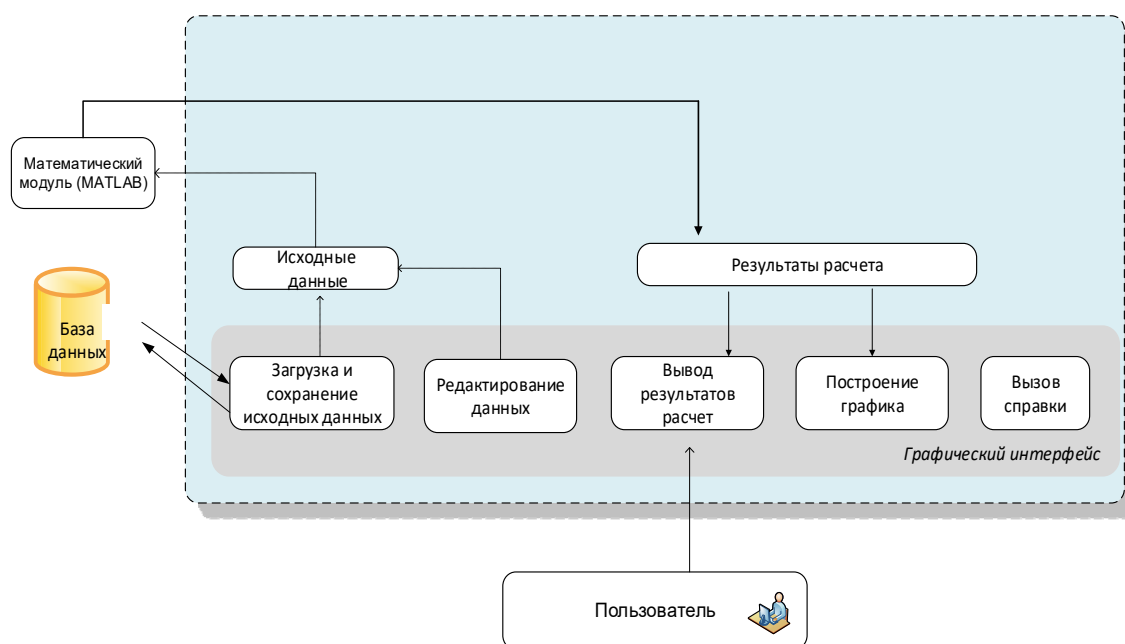


Рисунок 2.3 – Архитектура информационной системы

Каждый модуль выполняет определенные функции, которые описаны в таблице 2.1.

Таблица 2.1 – Функции модулей

Модуль	Функция
Загрузка и сохранение исходных данных	Извлечение данных из базы данных. Проверка корректности исходных данных
Редактирование исходных данных	Изменение данных. Проверка введенных данных на корректность
Математический модуль (MATLAB)	Загрузка данных в MATLAB. Запуск функции в Excel. Выгрузка данных из Excel.
Вывод результатов расчета	Представляет результаты расчета в численном виде.
Построение графика	Создание столбчатой диаграммы
Вызов справки	Предоставление справочной информации по работе с программой

2.3 Реализация модуля решения задачи оптимизации

Одним из инструментов для создания математических библиотек или внешних компонентов является Deployment Tool, представляющий собой графический интерфейс. Вызов данного инструмента осуществляется командой `deploytool`.

На рисунке 2.4 представлено стартовое окно Deployment Tool, на котором предлагается выбрать тип и создать новый проект.

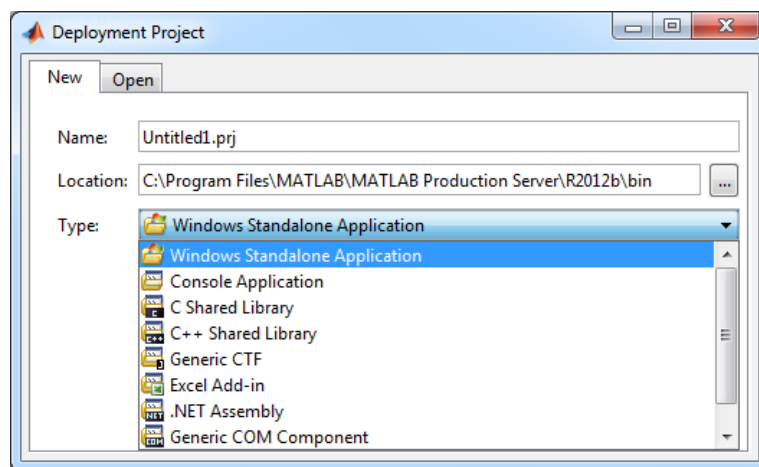


Рисунок 2.4 – Стартовое окно инструмента Deployment Tool в MATLAB

Название проекта определяет пространство имен, используемое в дальнейшем в Visual C#. Проект состоит из классов, которые в свою очередь включают М-файлы. Каждый М-файл включает одну функцию, которая вызывается как метод соответствующего класса в Visual C#.

Анализ нелинейности доменного процесса показал, что при решении задач оптимального распределения топливно-энергетических ресурсов, в силу относительно небольшого отклонения параметров от базового состояния, достаточно использовать линеаризованные зависимости.

Рассмотрим решение задачи линейного программирования в пакете MATLAB. Решение осуществляется с помощью функции `linprog`. Функция `linprog` решает задачу линейного программирования в форме:

$$f^T \cdot x \rightarrow \inf ,$$

$$A \cdot x \leq b, \quad (2.1)$$

$$lb \leq x \leq ub.$$

Основными входными данными функции linprog являются: вектор коэффициентов целевой функции f , матрица ограничений-неравенств A , вектор правых частей ограничений-неравенств b , матрица ограничений-равенств Aeq , вектор правых частей ограничений-равенств beq , вектор lb , ограничивающий план x снизу, вектор ub , ограничивающий план x сверху. На выходе функция linprog даёт оптимальный план x задачи (2.1) и экстремальное значение целевой функции $fval$.

Целевая функция примет вид:

$$Z(V_i) = \sum C_i \cdot V_i^{ПГ} \rightarrow \max, \quad (2.2)$$

$$V_i^{ПГ}$$

где $C_i = \alpha(e_i C_K - C_{ПГ}) + (1 - \alpha) \cdot C_{П}(\Delta\Pi_i^{ПГ} - e_i \Delta\Pi_i^K)$.

Здесь N – число печей в рассматриваемой группе (цехе); e_i – эквивалент замены кокса, кг кокса/(м³ ПГ); C_K – стоимость кокса, руб/кг; $C_{ПГ}$ – стоимость природного газа, руб/м³; $C_{П}$ – условно постоянный коэффициент, руб/т чугуна (принять $C_{П}=1$); $\Delta\Pi_i^{ПГ}$ – прирост производства чугуна при изменении расхода природного газа на i -й печи, т чугуна/(м³ ПГ); $\Delta\Pi_i^K$ – прирост производства чугуна при изменении расхода кокса на i -й печи, т чугуна/(кг кокса); $V_i^{ПГ}$ – расход природного газа на i -й доменной печи, м³/ч.

Матрица ограничений неравенств A включает в себя ограничения в целом по цеху, а именно:

По расходу природного газа:

$$\sum a_{1i} \cdot V_i^{ПГ} \leq b_1, \text{ где } a_{1i} = 1, b_1 = V_{ПГ}^{\Sigma}$$

$V_{\Sigma}^{ПГ}$ – резерв по расходу природного газа в целом по цеху, м³/ч.

По расходу кокса:

$$\sum a_{2i} \cdot V_i^{ПГ} \leq b_2, \text{ где } a_{2i} = 0,001 \cdot e_i, b_2 = \sum [K_{0i} + 0,001 \cdot e_i \cdot V_{i,0}^{ПГ}] - K_{\Sigma}$$

K_{0i} – расход кокса на i -ю печь в базовом периоде, т/ч; K_{Σ} – запасы кокса по цеху, т/ч; $V_{i0}^{ПГ}$ – расход природного газа на i -й печи в базовом периоде, м³/ч.

По объему производства чугуна:

$\sum a_{3i} * V_i^{ПГ} \geq b_3$, где $a_{3i} = \Delta\Pi_i^{ПГ} - e_1 * \Delta\Pi_i^K$ $b_3 = \Pi_{\Sigma} + V_{i0}^{ПГ} * (\Delta\Pi_i^{ПГ} - e_i * \Delta\Pi_i^K) - \Pi_{i0}$
 Π_{Σ} – требуемое производство чугуна в цехе, т/ч; Π_{i0} – производительность i -й печи по чугуну, т/ч.

Аналогичным образом формируются расчетные формулы для технологических ограничений на каждую из печей цеха, такие как обобщённый показатель низа печи, теоретическая температура горения, содержание кремния в чугуне, отношение теплоемкостей поток шихты и газа в верхней ступени теплообмена, учет полноты конверсии природного газа и газодинамического режима доменной плавки и содержание серы в чугуне.

Для использования внешней библиотеки среда разработки Visual Studio 2015 и язык программирования Visual C# предусматривает механизм ссылок на сборку. Для подключения написанных в MATLAB функций необходимо добавить ссылку на созданную библиотеку, а также на библиотеку MWArray, находящуюся в папке установленной программы по адресу “C:\Program Files\MATLAB\MATLABProductionServer\R2012b\toolbox\dotnetbuilder\bin\win32\v2.0\MWArray.dll”. На рисунке 2.5 представлено окно добавления новой ссылки на сборку.

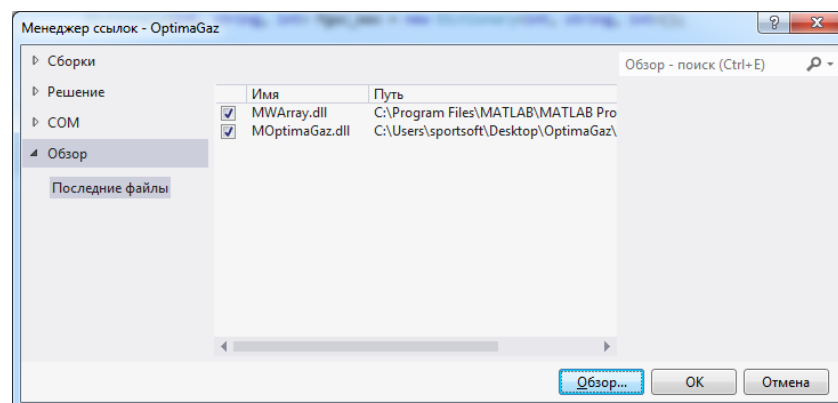


Рисунок 2.5 – Окно добавления ссылки на сборку в Visual Studio

Библиотека *MWArray* предназначена для обмена данными между средой программирования MATLAB и платформой .NET. Она обеспечивает

преобразование типов данных, отслеживание ошибок выполнения кода, экспорт графики в приложения ASP.NET и др.

Отметим обязательность соблюдения разрядности MATLAB и создаваемого программного обеспечения. Также замечена нестабильность работы при работе с 64-разрядной архитектурой, поэтому при разработке использовалась 32-разрядная версия MATLAB.

Для использования библиотеки необходимо добавить описание пространства имен:

```
using MathWorks.MATLAB.NET.Utility;  
using MathWorks.MATLAB.NET.Arrays;  
using MOptimaGaz;
```

Ниже приведен листинг программы с комментариями, вызывающий внешнюю функцию MATLAB и выводящий результаты решения задачи.

```
MClassOptimaGaz MObject = new MClassOptimaGaz();  
MWNumericArray descriptor = null; // Массив возвращаемого параметра  
int x = 8; // Количество неизвестных задачи оптимизации  
double[] lb = new double[x];  
double[] ub = new double[x];  
double[] f = new double[x];  
double[,] Aeq = new double[3, x]; // Всего 3 ограничения-неравенства  
double[] beq = new double[3]; // Всего 3 ограничения-неравенства  
// Присваивание переменных lb, ub, Aeq, beq и f реальных значений  
MWArray[] result = MObject.MOptimaGaz(2, (MWNumericArray)f,  
(MWNumericArray)Aeq, (MWNumericArray)beq, (MWNumericArray)lb,  
(MWNumericArray)ub);  
// Первый параметр - это количество возвращаемых аргументов  
for (int j = 0; j < result.Length; j++)  
{  
    MessageBox.Show("Вывод массива " + j.ToString());  
    descriptor = (MWNumericArray)result[j];
```

```

double[,] d_descriptor =
(double[,])descriptor.ToArray(MWArrayComponent.Real);
// Преобразование массива MWNumericArray к массиву типа double
for (int i = 0; i < d_descriptor.Length; i++)
{
    MessageBox.Show(d_descriptor[i, 0].ToString());
}
}

```

Из программы опущен блок, присваивающий переменным *lb*, *ub*, *Aeq*, *beq* и *f* значения. Если решение не найдено, то результат содержит пустой массив.

2.4 Разработка базы данных

Процесс построения инфологической модели базы данных реализован с использованием CASE-методологии проектирования [18–21], которая включает в себя следующие этапы:

- 1) определение сущностей;
- 2) определение связей между сущностями;
- 3) задание первичных и составных ключей;
- 4) приведение модели к требуемому уровню нормальной формы;
- 5) переход к физическому описанию модели: назначение соответствий имя сущности – имя таблицы, атрибут сущности – атрибут таблицы; задание ограничений предметной области;
- 6) генерация базы данных, т.е. формирование физической схемы для конкретной СУБД.

В ходе инфологического моделирования учтены следующие общие требования к базе данных: централизованное хранение данных; обеспечение безопасности и надежности данных; обеспечение целостности и согласованности данных.

В разработанной базе данных должны храниться значения технологических показателей работы доменных печей за прошедший период времени, которые необходимы для решения задачи оптимального распределения топливно-энергетических ресурсов. В модели выделены сущности, представленные на рисунке 2.6. Описание атрибутов информационных сущностей базы данных системы приведено в таблице 2.2–2.8.

Таблица 2.2 – Характеристика сущностей базы данных

Наименование таблицы	Описание	Примечание
t_sprav_indicators	Название всех показателей	Ключевое поле: Код показателя
t_sprav_koefs	Название групп коэффициентов модели	Ключевое поле: Код коэффициента
t_sprav_pechi_limits	Название всех ограничений на печи	Ключевое поле: Код ограничения на печь
t_sprav_pechi	Данные о количество печей в доменном цехе	Ключевое поле: Код доменной печи
t_data_indicators	Значение показателей доменных печей	Ключевое поле: Код доменной печи, код индикатора, значение
t_data_koefs	Значение коэффициентов модели	Ключевое поле: Код доменной печи, код коэффициента, значение
t_data_pechi_limits	Значение ограничений для каждой доменной печи	Ключевое поле: Код доменной печи, код ограничения, значение

Таблица 2.3 – Спецификация сущности t_sprav_indicators

Наименование атрибута	Описание атрибута	Тип данных	Примечание
indicator_id	Идентификатор показателя	int	первичный ключ
symbol	Символьное название показателя для расчета	nvarchar(20)	
name	Название показателя	nvarchar(255)	

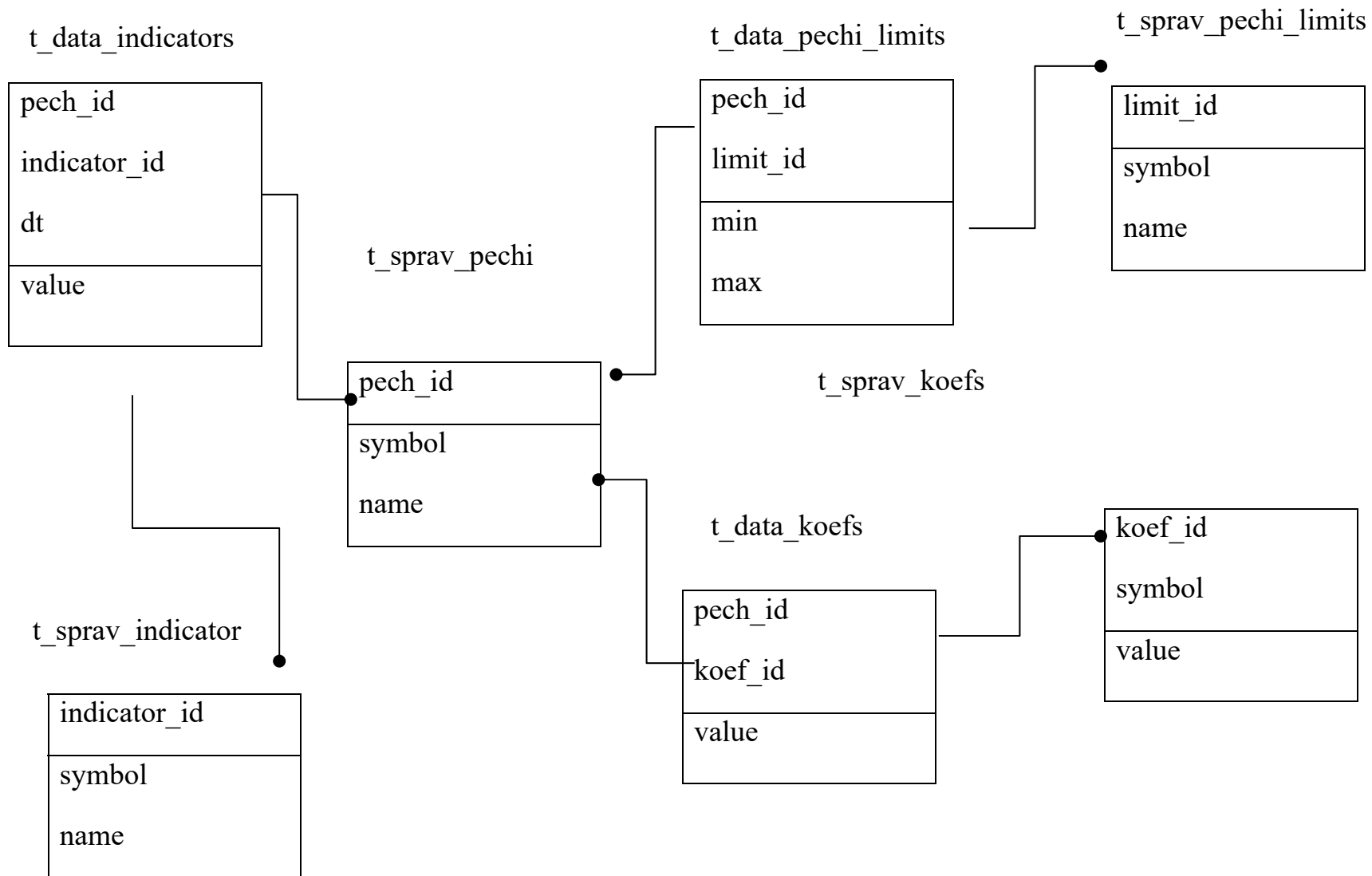


Рисунок 2.6 – Инфологическая модель базы данных

Таблица 2.4 – Спецификация сущности t_sprav_pechi

Наименование атрибута	Описание атрибута	Тип данных	Примечание
pech_id	Идентификатор печи	int	первичный ключ
name	Название печи	nvarchar(255)	

Таблица 2.5 – Спецификация сущности t_sprav_pechi_limits

Наименование атрибута	Описание атрибута	Тип данных	Примечание
limit_id	Идентификатор печи	int	первичный ключ
symbol	Символьное название ограничения	nvarchar(20)	
name	Название ограничения	nvarchar(255)	

Таблица 2.6 – Спецификация сущности t_data_indicators

Наименование атрибута	Описание атрибута	Тип данных	Примечание
pech_id	Идентификатор печи	int	первичный ключ
indicator_id	Идентификатор показателя	int	первичный ключ
dt	Дата	smalldatetime	первичный ключ
value	Значение показателя	decimal(17,10)	

Таблица 2.7 – Спецификация сущности t_data_koefs

Наименование атрибута	Описание атрибута	Тип данных	Примечание
koef_id	Идентификатор коэффициента	int	первичный ключ
pech_id	Идентификатор печи	int	первичный ключ
value	Значение коэффициента	decimal(17,10)	

Таблица 2.8 – Спецификация сущности t_data_pechi_limits

Наименование атрибута	Описание атрибута	Тип данных	Примечание
pech_id	Идентификатор печи	int	первичный ключ
limit_id	Идентификатор ограничения	int	первичный ключ
min	Минимальное значение коэффициента	decimal(14,4)	
max	Максимальное значение коэффициента	decimal(14,4)	

После создания инфологической модели системы переходим к созданию даталогической модели на основе конкретной СУБД. Для реляционной модели данных даталогическая модель – набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи [14, 16].

База данных разработана в среде Microsoft SQL Server 2008 R2 [18]. Среда управления СУБД SQL Server Management Studio – это набор административных средств для управления компонентами, принадлежащих SQL Server. Эта интегрированная среда позволяет пользователям выполнять разнообразные задачи, например резервное копирование данных, редактирование запросов и автоматизацию общих функций в одном интерфейсе.

Преобразование инфологической модели в даталогическую модель выбранной реляционной СУБД осуществляется в следующей последовательности.

1. Для каждой сильной сущности ER-модели создается отдельная таблица, а для каждого атрибута сущности создается столбец таблицы. Ключевой атрибут становится первичным ключом, а дополнительные ключевые атрибуты - потенциальными ключами.

2. Для каждой слабой сущности также создается отдельная таблица, в которой должны присутствовать ключевые столбцы доминирующих таблиц. В зависимости от вида связи устанавливаются ключевые атрибуты таблицы.

3. Далее необходимо создать внешние ключи, обеспечивающие ссылочную целостность, по указанному типу связи в ER-модели.

В физической модели каждой сущности будет соответствовать таблица базы данных, а каждому атрибуту – поле таблицы.

После создания таблиц и ключевых полей создаем связи (рисунок 2.7).

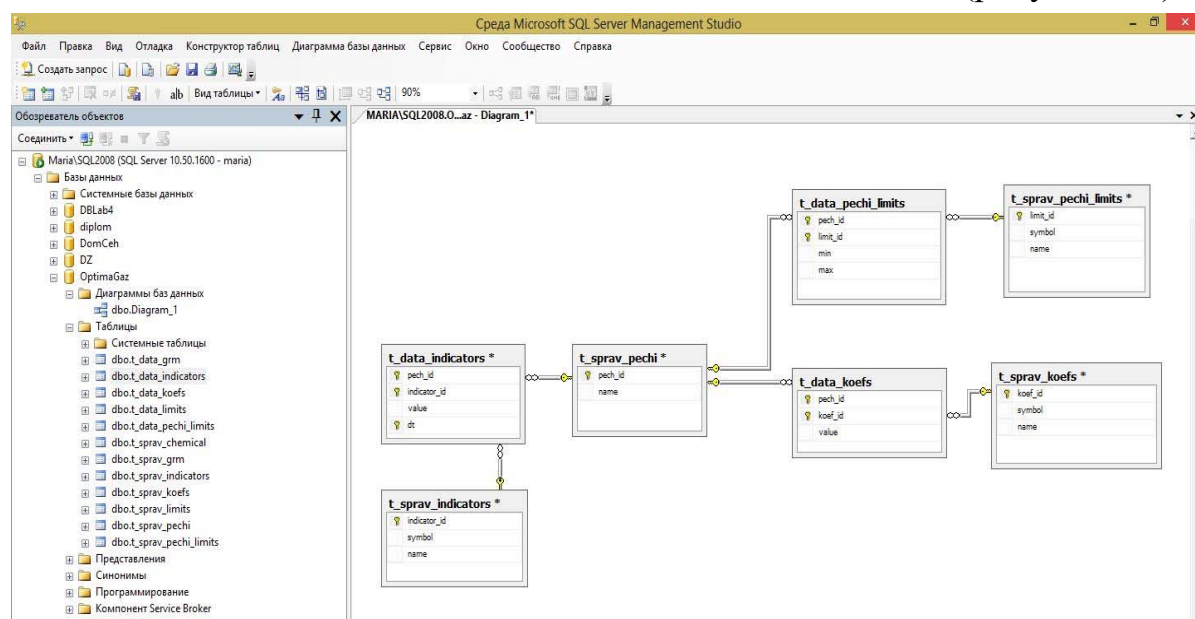


Рисунок 2.7 – Даталогическая модель схема базы данных

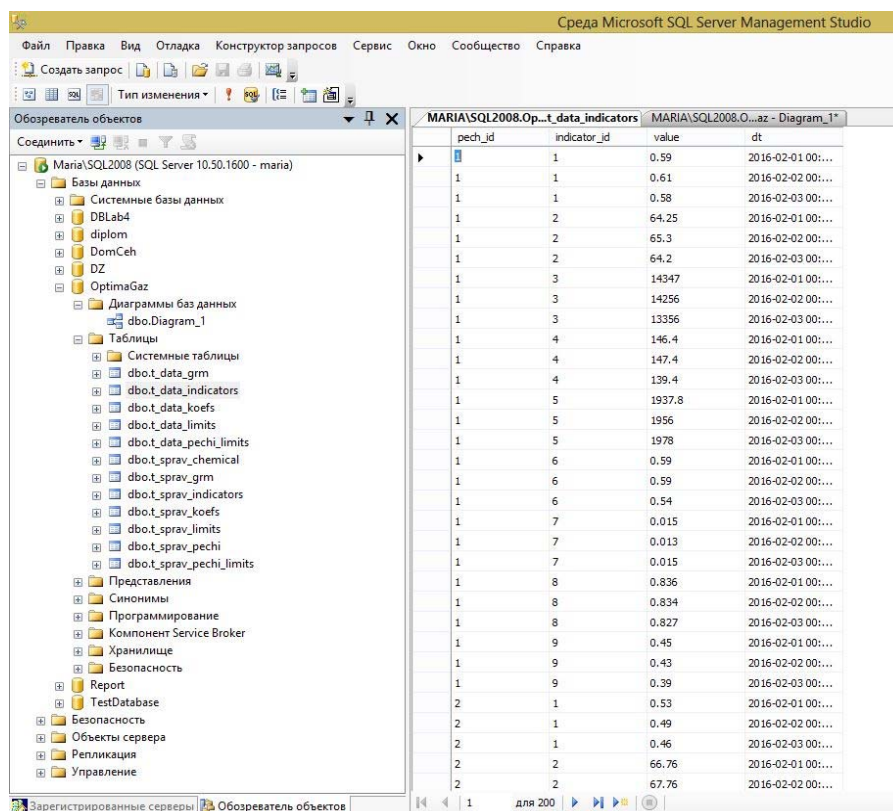
После создания связей между сущностями наполняем базу данных необходимыми данными. На рисунке 2.8 приведены таблицы с данными.

Для удобства передачи данных между базой данных и пользовательским приложением необходимо использовать хранимые процедуры.

Хранимые процедуры представляют собой группы связанных между собой операторов SQL, применение которых делает работу программиста более легкой и гибкой, поскольку выполнить хранимую процедуру часто оказывается гораздо проще, чем последовательность отдельных операторов SQL.

Выполнение в базе данных хранимых процедур вместо отдельных операторов SQL дает пользователю следующие преимущества:

- необходимые операторы уже содержатся в базе данных;
- все они прошли этап синтаксического анализа и находятся в исполняемом формате; перед выполнением хранимой процедуры SQL Server генерирует для нее план исполнения, выполняет ее оптимизацию и компиляцию;
- хранимые процедуры поддерживают модульное программирование, так как позволяют разбивать большие задачи на самостоятельные, более мелкие и удобные в управлении части;
- хранимые процедуры могут вызывать другие хранимые процедуры и функции;
- хранимые процедуры могут быть вызваны из прикладных программ других типов.



pech_id	indicator_id	value	dt
1	1	0.59	2016-02-01 00:...
1	1	0.61	2016-02-02 00:...
1	1	0.58	2016-02-03 00:...
1	2	64.25	2016-02-01 00:...
1	2	65.3	2016-02-02 00:...
1	2	64.2	2016-02-03 00:...
1	3	14347	2016-02-01 00:...
1	3	14256	2016-02-02 00:...
1	3	13356	2016-02-03 00:...
1	4	146.4	2016-02-01 00:...
1	4	147.4	2016-02-02 00:...
1	4	139.4	2016-02-03 00:...
1	5	1937.8	2016-02-01 00:...
1	5	1956	2016-02-02 00:...
1	5	1978	2016-02-03 00:...
1	6	0.59	2016-02-01 00:...
1	6	0.59	2016-02-02 00:...
1	6	0.54	2016-02-03 00:...
1	7	0.015	2016-02-01 00:...
1	7	0.013	2016-02-02 00:...
1	7	0.015	2016-02-03 00:...
1	8	0.836	2016-02-01 00:...
1	8	0.834	2016-02-02 00:...
1	8	0.827	2016-02-03 00:...
1	9	0.45	2016-02-01 00:...
1	9	0.43	2016-02-02 00:...
1	9	0.39	2016-02-03 00:...
2	1	0.53	2016-02-01 00:...
2	1	0.49	2016-02-02 00:...
2	1	0.46	2016-02-03 00:...
2	2	66.76	2016-02-01 00:...
2	2	67.76	2016-02-02 00:...

Рисунок 2.8 – Таблица t_data_indicators

Хранение процедур в том же месте, где они исполняются, обеспечивает уменьшение объема передаваемых по сети данных и повышает общую производительность системы [19].

На рисунке 2.9 показана диаграмма взаимодействия объектов при работе хранимой процедуры dbo.ex_SEL_Indicators.

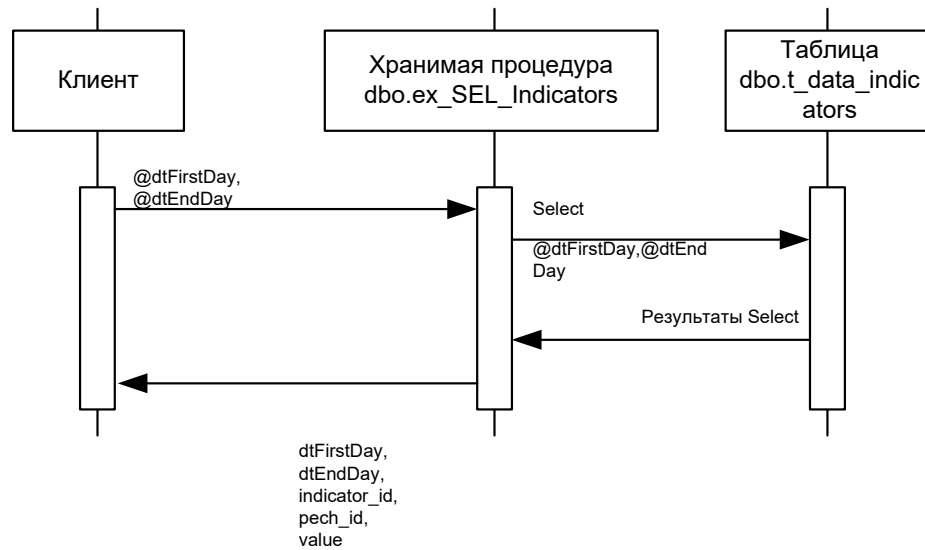


Рисунок 2.9 – Диаграмма взаимодействия выполнения хранимой процедуры `dbo.ex_SEL_Indicators`

Листинг хранимой процедуры `dbo.ex_SEL_Indicators` приведен ниже.

```

SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS OFF
GO

```

/*-----

Проект "OptimaGaz"

Пример обработки базы данных MS SQL Server 2008 R2

Разработчик: Бякова Мария Андреевна

Функции:

Таблица "Показатели работы доменных печей по датам "

Чтение данных из таблицы

Входные данные:

`@dtFirstDay` – первый день отчетной даты

`@dtFirstDay` – второй день отчетной даты

Пример вызова:

```
EXEC dbo.ex_SEL_Indicators @dtFirstDay=  
*/-----
```

```
CREATE PROC dbo.ex_SEL_Indicators  
    @dtFirstDay smalldatetime  
    @dtEndDay smalldatetime  
AS
```

SET NOCOUNT ON -- не передавать сообщения о количестве обработанных записей

```
SELECT F.dt,  
       F.id_pechi,  
       F.id_indicator,  
       F.dt,  
       F.vallue,  
FROM dbo.t_data_indicators AS F  
WHERE      dt      BETWEEN      '"+dateStart+"T00:00:00.000'      AND  
 '"+dateEnd+"T00:00:00.000' " +  
          "GROUP BY indicator_id, pech_id ORDER BY indicator_id ASC,  
 pech_id ASC"  
GO  
SET QUOTED_IDENTIFIER OFF  
GO  
SET ANSI_NULLS ON  
GO
```

Все СУБД представляют доступ к информации базы данных только зарегистрированным пользователям, имеющим учетные записи этой базы. Для создания учетной записи каждого нового пользователя требуется создать стандартный объект USER с обязательным указанием пароля данного пользователя. Для этого на вкладке Безопасность выберем пункт «Создать» (рисунок 2.10).

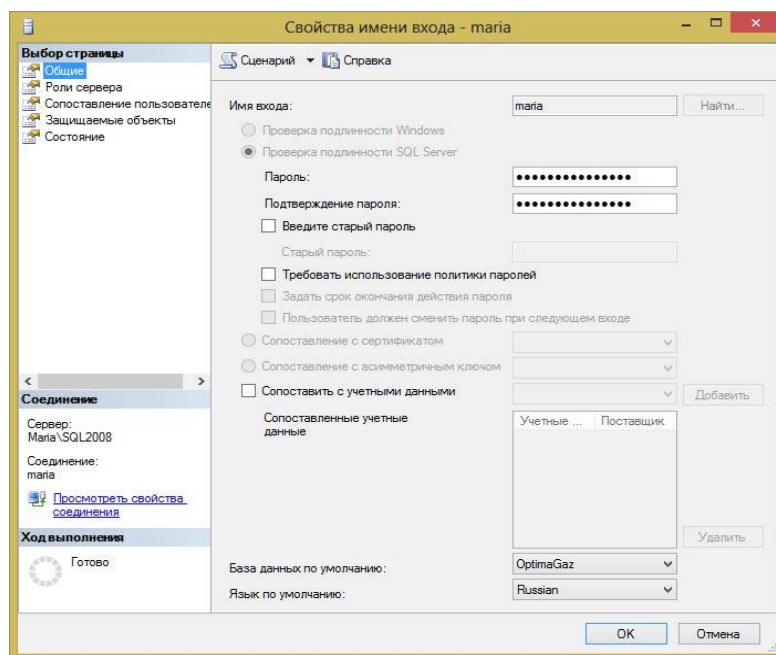


Рисунок 2.10 – Создание нового пользователя

Для сопровождения базы данных необходимо разработать программное обеспечение.

2.5 Создание программного обеспечения в системе управления версиями Atlassian Bitbucket

Для работы с Atlassian Bitbucket необходимо установить TortoiseGit [20-21]. Это визуальный клиент системы управления исходными кодами программ git для ОС Microsoft Windows.

Репозитории объединяются в проекты. Поэтому для создания репозитория прежде всего нужно создать проект. Для этого создаем проект в разделе Projects. Переходим в него и создаем репозиторий (рисунок 2.11).

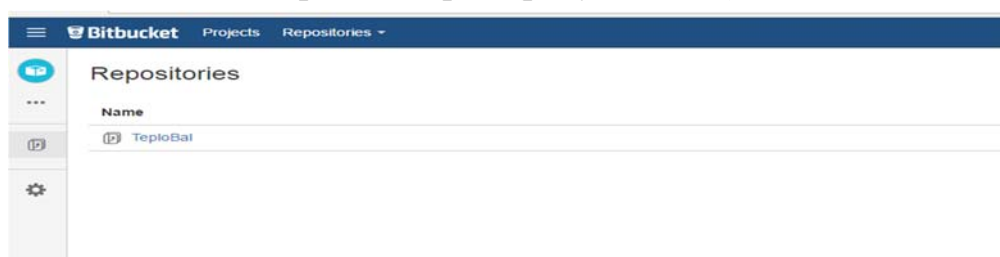


Рисунок 2.11 – Создание репозитория

Далее создадим репозиторий «Git Create Repository here». Еще раз правой кнопкой и переходим в настройки репозитория, чтобы настроить соединение (рисунок 2.12).

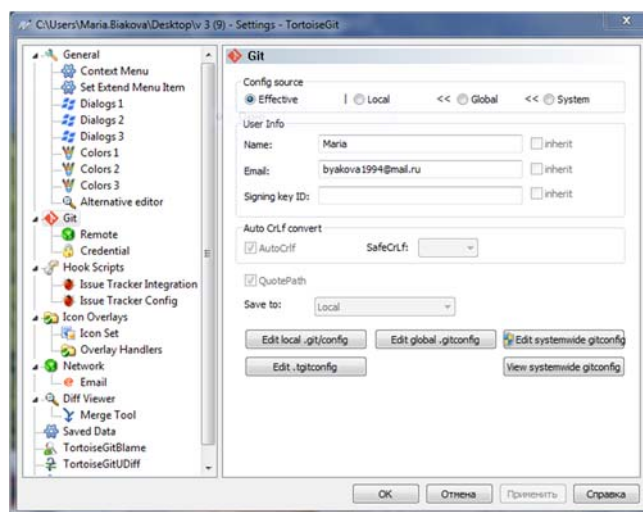


Рисунок 2.12 – Настройка соединения

Наполняем репозиторий файлами проекта (рисунок 2.13). Выполняем Коммит и Пуш запрос. При выполнении коммита обязательно заполнение поля «Сообщение».

При запросе пароля требуется ввести пароль от своего аккаунта в bitbucket. На сервере можно посмотреть историю изменений и файлы проекта.



Рисунок 2.13 – Наполнение репозитория файлами проекта

2.6 Разработка пользовательского интерфейса

Программное обеспечение разработано в среде Microsoft Visual Studio 2013 на языке C# [22, 23].

Microsoft Visual Studio 2013 – это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского

интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию. Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки (IDE) Visual Studio.

C# – объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML [24, 25].

Для создания диаграммы использована открытая программная библиотека ZedGraph, которая находится в свободном доступе в интернете [24].

Перед непосредственной разработкой программного продукта был создан эскиз интерфейса пользователя. На рисунке 2.14 приведена главная форма программы, через которую будет производиться основное взаимодействие с пользователем. На данной форме пользователь видит значения производственных показателей работы доменных печей, так же может вызывать требуемые модули программы.

При работе с программой могут возникнуть ситуации, которые отрицательно влияют на ее работу и даже приводят к «зависанию». Чтобы избежать подобных исключительных ситуаций в программе предусмотрены процедуры их обработки.

В программе для расчета используются только численные величины, поэтому важной задачей является не допустить ввода пользователем, случайно

или преднамеренно любые другие символы в поля с исходными значениями. При попытке ввода любой некорректной последовательности символов, пользователь не сможет ввести их. Корректными являются любые целые числа и числа с дробной частью.

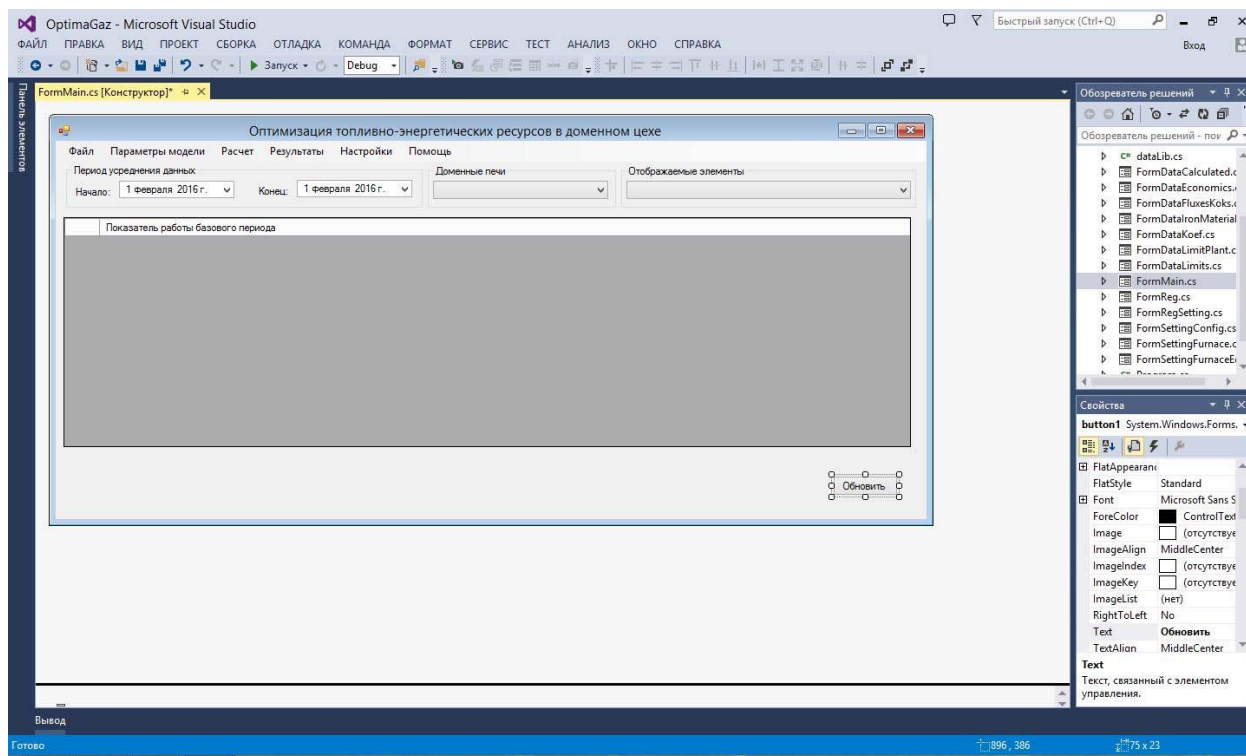


Рисунок 2.14 – Эскиз главной формы программного средства

Также во избежание ошибок пользователю недоступна кнопка «Результаты» до тех пор, пока не будет произведен расчет и найдены результаты.

Справка разработана в пакете Help&Manual 5. С помощью программы Help&Manual 5 можно создавать документацию в форматах CHM, HTML, PDF, RTF. В создаваемый файл справки можно добавить оглавление, указатель, поиск (в тех форматах, где это поддерживается).

Вызвать справку в программе можно двумя способами:

- нажав F1 в любой части программы;
- нажав кнопку «Справка» на форме.

На рисунке 2.15 представлено окно редактора Help&Manual 5.

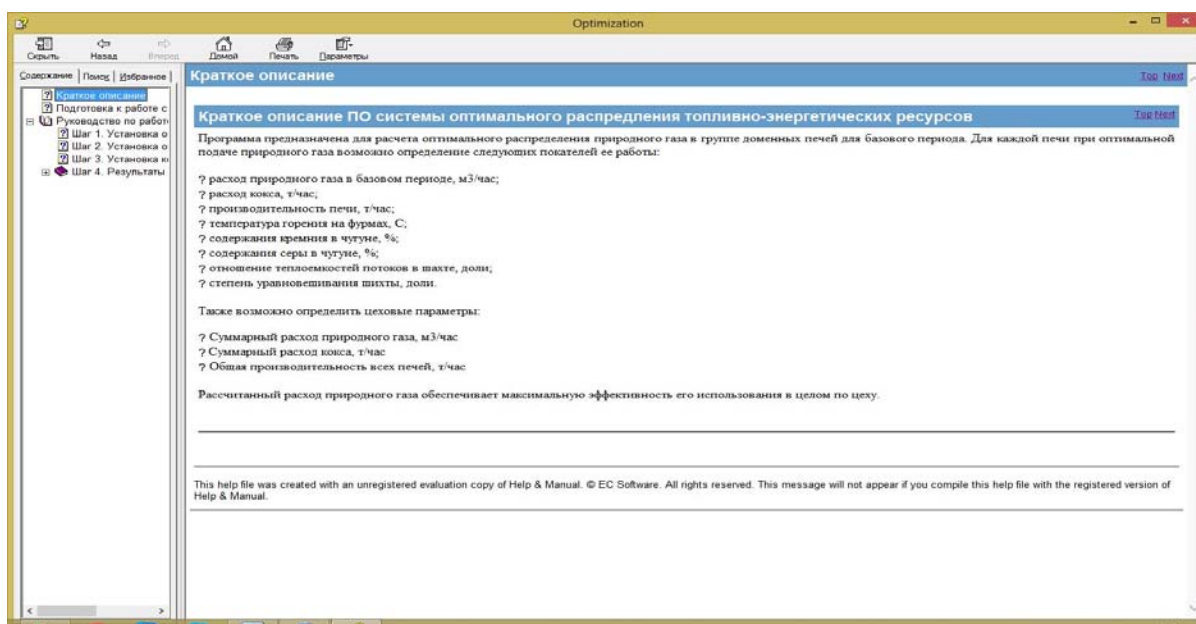


Рисунок 2.15 – Главная страница справки

Создание дистрибутива – неотъемлемая часть разработки ПО. Дистрибутив используется для распространения программного продукта. Дистрибутив должен быть интуитивно понятным пользователю. При установке программы должен быть указан путь по умолчанию, по которому установится программа (обычно это Program Files), но у пользователя должна быть возможность при необходимости изменить путь. Также пользователь должен видеть ход установки, а по окончании установки иметь возможность создать ярлык программы на рабочем столе и запустить программу при выходе из инсталлятора.

Для создания дистрибутива была использована возможности Visual Studio 2015 InstallShield. InstallShield – система создания инсталляторов и пакетов программного обеспечения для Microsoft Windows и Linux с закрытыми исходными кодами. Вид главного окна программы приведен на рисунке 2.16.

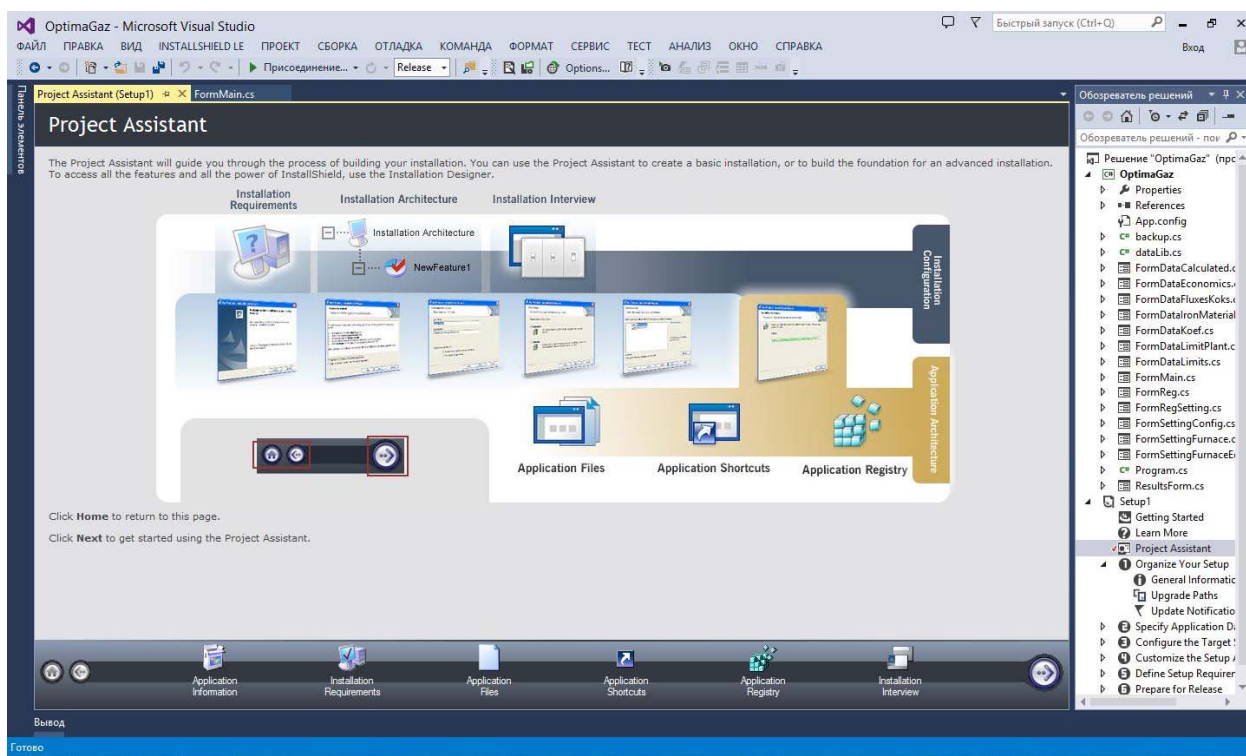


Рисунок 2.16 – Вид главного окна InstallShield

2.7 Вывод по главе

При проектировании и разработки программного обеспечения системы оптимального распределения топливно-энергетических ресурсов, используется подход интеграции языка программирования Visual C# и пакета MATLAB. Данный подход позволяет разрабатывать стильные и удобные пользовательские интерфейсы в сочетании с мощным инструментом для решения математических задач.

3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Установка и настройка программного продукта

Чтобы использовать программу, пользователь должен установить ее на свой компьютер. Для этого пользователю следует запустить на выполнение установочный файл программы setup.exe.

Программа не запустится, если на компьютере не установлен .NET Framework 4.

3.2 Функциональные возможности программного продукта

При проектировании программы была поставлена задача достижения определенной функциональности. По окончании создания программного средства достигнуты следующие функции:

- возможность загрузки данных из базы данных и корректировки значений данных, для достижения универсальности применения программного средства;
- программный контроль за процессом корректировки значений исходных величин для исключения некорректной работы программы;
- представление результатов расчета в численном и графическом виде.

Таким образом, заложенная в начале этапа проектирования функциональность была достигнута.

3.3 Описание работы программного обеспечения

Запуск программы осуществляется двойным щелчком левой кнопкой мыши на файле с именем Optimpg.exe. После этого на экране появляется главная форма приложения (рисунок 3.1). На главной форме отображены показатели работы доменных печей, которые необходимы для решения задачи оптимизации.

Оптимизация топливно-энергетических ресурсов в доменном цехе

Файл Параметры модели Расчет Результаты Настройки Помощь

Период усреднения данных: Начало: 1 февраля 2016 г. Конец: 3 февраля 2016 г.

Доменные печи: ДП-1, ДП-2, ДП-3, ДП-4, ДП-5, ДП-6

Отображаемые элементы: Технологические показатели для расчета

Показатель работы базового периода	ДП-1	ДП-2	ДП-3	ДП-4	ДП-5	ДП-6	ДП-7	ДП-8
Отношение тепл...	0.835	0.8767	0.8035	0.84675	0.8345	0.84665	0.8725	0.8904
Производительн...	146.2	136.4	134.15	122.15	138.45	138.8	191.4	151.6
Расход кокса на ...	64.175	66.73	56.07	49.76	62.935	60.015	81.67	69.7
Расход природно...	14346.5	15229.5	11822	12527.5	11305	11005	17449	15196.5
Содержание кре...	0.585	0.51	0.655	0.6	0.533	0.435	0.4	0.6
Содержание сер...	0.015	0.014	0.013	0.014	0.017	0.016	0.013	0.014
Степень уравни...	0.45	0.451	0.4525	0.46	0.4555	0.458	0.4565	0.4635
Теоретическая т...	1975.5	1957.5	2090.35	1989.25	1997	1925	1974	1981.5
Эквивалент зам...	0.585	0.52	0.845	0.59	0.745	0.79	0.87	0.775

Обновить

Рисунок 3.1 – Главная форма программного средства

Выбор подпункта меню «Ограничения на цех» во вкладке «Параметры модели» приводит к появлению окна, вид которого представлен на рисунке 3.2. Здесь отображены суммарные резервные величины по расходам природного газа и кокса в целом по цеху, а также минимально допустимая (требуемая) производительность цеха по чугуна. Пользователь в любой момент может скорректировать эти величины.

Ограничения на цех

Здесь устанавливаются общие ресурсы и требуемое производство чугуна на группу доменных печей

Резерв по природному газу, м3/час	Базовый режим	114325	Закрыть
Резерв по расходу кокса, т/час		1217	
Требуемое производство чугуна, т/час		15096	

Отмена

Рисунок 3.2 – Окно ввода и корректировки ограничений на цех

Команда «Ограничения на печи» вызывает появление на экране окна (рисунок 3.3), в котором пользователь может определить различные типы

ограничений, а также изменить количественные величины каждого типа для каждой отдельно выбранной для расчета доменной печи.

В программе возможен учет следующих типов ограничений на доменную печь с установленными максимальными и минимальными значениями соответствующих параметров:

- прямой расход природного газа;
- теоретическая температура горения;
- степень уравнивания шихты газовым потоком;
- содержание серы в чугуна;
- содержание кремния в чугуна;
- показатель теплового состояния низа печи.

С помощью флажков пользователь может установить или исключить учет соответствующего типа ограничений.

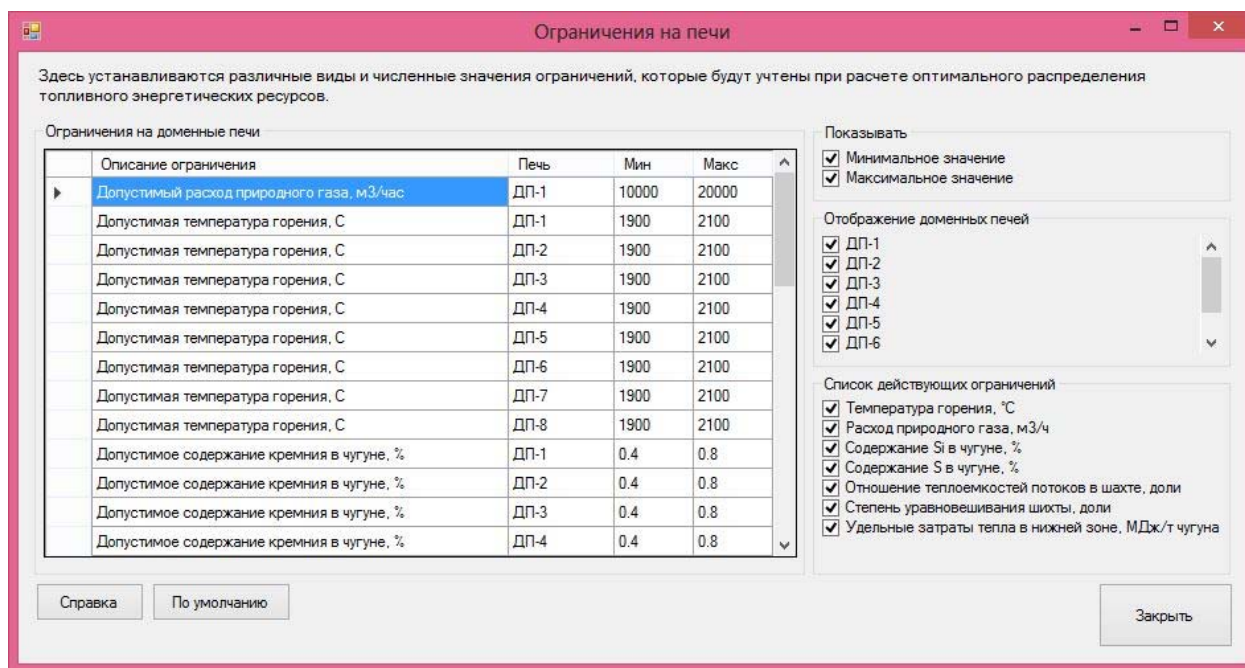


Рисунок 3.3– Окно установки ограничений на каждую доменную печь

Команда «Коэффициенты» позволяет пользователю просмотреть и скорректировать численные значения коэффициентов для каждой печи, которые используются в оптимизационной модели. Вид диалогового окна представлен на рисунке 3.4. При наведении курсора мыши на поле редактирования в верхней

части окна появляется пояснение физического смысла соответствующего коэффициента.

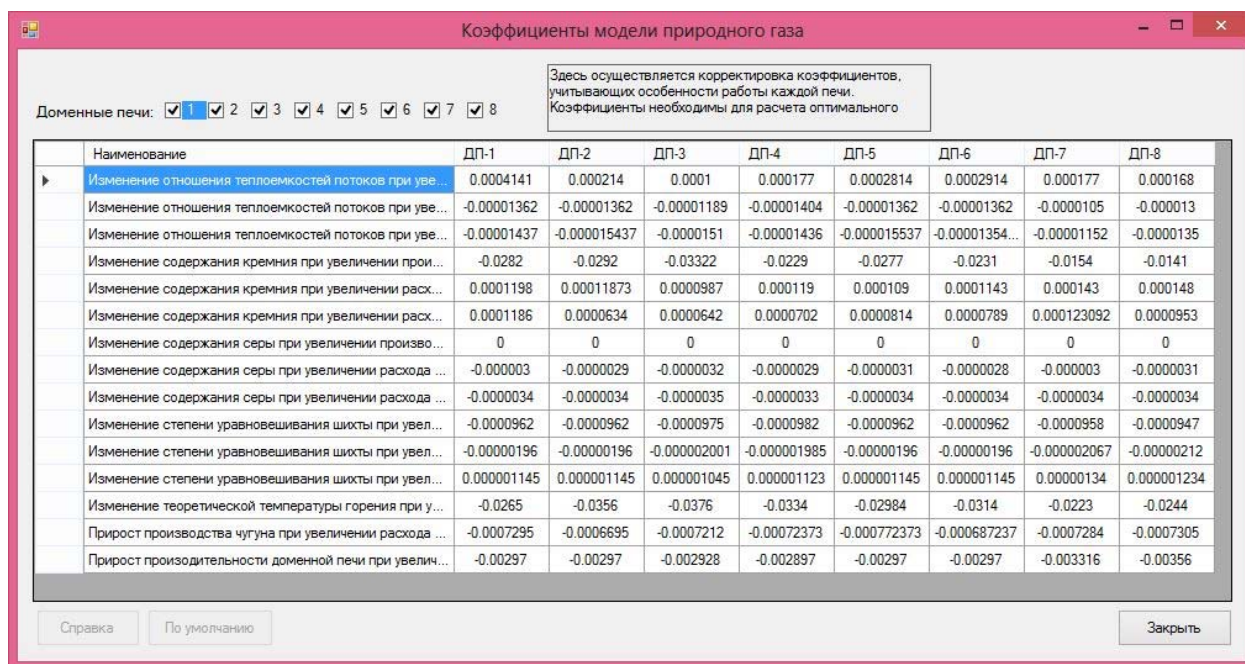


Рисунок 3.4 – Окно для отображения и корректировки коэффициентов

Команда «*Экономические параметры*» выводит на экран окно, в котором пользователь может просмотреть и скорректировать стоимостные характеристики природного газа и кокса (рисунок 3.5).

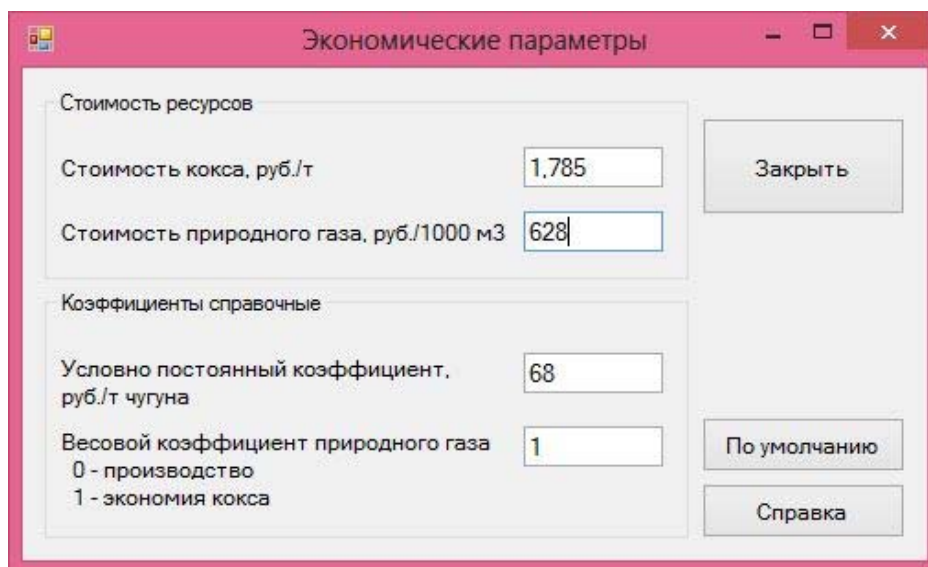
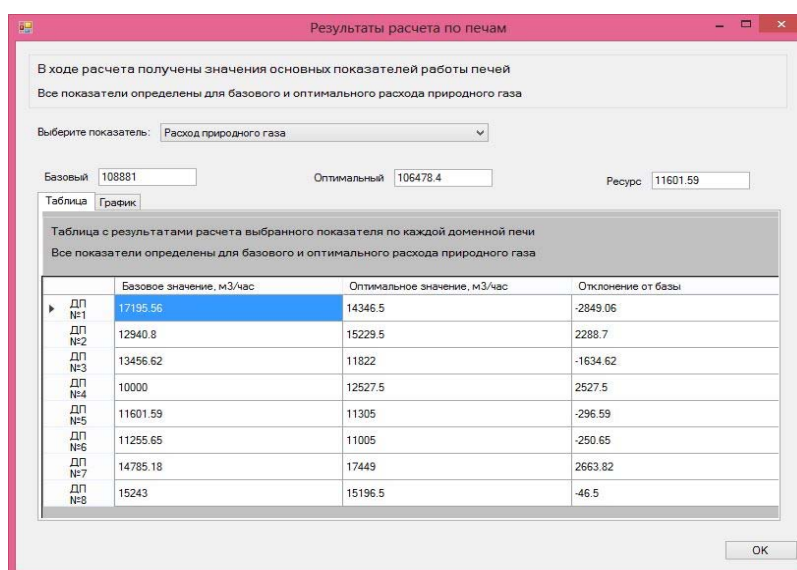


Рисунок 3.5 – Окно ввода стоимостных характеристик

Для того, чтобы произвести расчет необходимо выбрать команду «Расчет» в главном меню программы. Расчет выполняется на основе усредненных базовых показателей работы доменных печей за прошедший период работы с учетом выбранных типов ограничений. В случае успешного выполнения расчета пользователю на экран выводится специальное сообщение.

После этого в главном меню программы активизируется команда «Результаты», которая недоступна пользователю без выполнения команды меню «Расчет». Команды подменю разделены на следующие функциональные категории (рисунок 3.6):

а



б

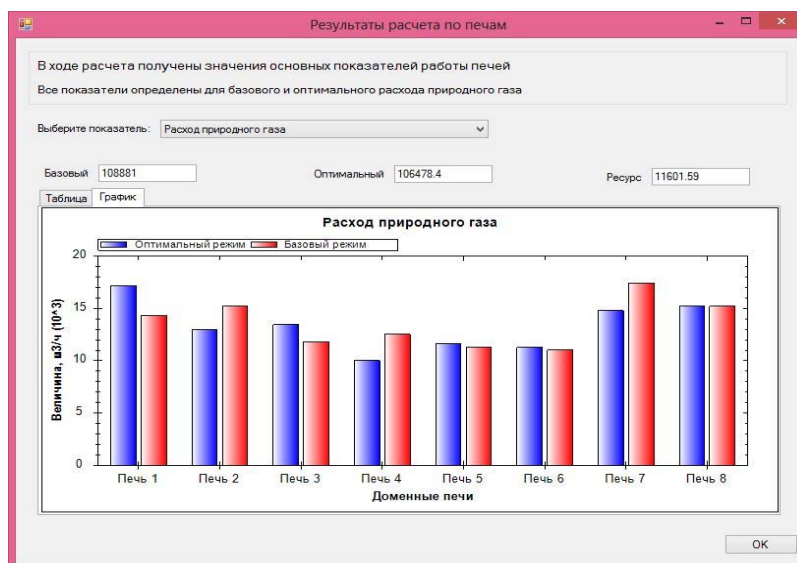


Рисунок 3.6 – Окно отображения результатов расчета:

а – табличное отображение, *б* – графическое отображение

– подпункт меню «*Таблица по цеху*» позволяет отобразить в табличном виде значения основных показателей работы всех доменных печей, участвующих в расчете. Отображение осуществляется в специальном окне, вид которого представлен на рисунке 3.6,а. Здесь представлены оптимальные и базовые значения выбранного показателя для каждой доменной печи, а также относительное отклонение. Пользователь может выбрать из списка любой доступный показатель, и содержимое окна автоматически обновится;

– подпункт меню «*Диаграмма*» вызывает появление окна с графической интерпретацией полученных результатов в виде диаграммы (рисунок 3.6,б). Столбики диаграммы отображают для каждой печи базовое и оптимальное значения показателя, который выбран для показа. Пользователь может перемещаться по закладкам «Таблица» и «Диаграмма», чтобы пересматривать результаты в удобном для него виде.

3.4 Вывод по главе

Реализованное программное обеспечение обладает необходимым функционалом для работы, а также имеет удобный и интуитивно понятный интерфейс.

ЗАКЛЮЧЕНИЕ

В процессе проектирования и создания информационно-моделирующей системы «Расчет оптимального распределения природного газа в группе доменных печей для обеспечения максимальной эффективности его вдувания» было создано следующее:

- функциональная модель проектируемой программы;
- тестовый файл расчета и проверки методики расчета Excel;
- программное средство для расчета оптимального распределения природного газа в группе доменных печей в базовом периоде;
- создана база данных для хранения данных для расчета;
- справочная система программы;
- руководство пользователя.

Разработанное программное средство отвечает всем задачам, определенным в начале проектирования, обеспечивает заданную функциональность.

Информационная система ориентирована на использование специалистами инженерно-технологической группы доменных цехов металлургических предприятий. Использование системы помогает пользователям оценить, насколько эффективно был использован природный газ. В частности, можно рассчитать для прошедшего периода оптимальный расход природного газа на каждую печь, определить показатели работы печей при этом расходе и выполнить сравнительный анализ всех вышеперечисленных показателей при базовой и оптимальной подаче природного газа.

Она может быть использована в учебном процессе для обучения бакалавров и магистрантов по направлениям «Металлургия» и «Информационные системы и технологии».

Работа была представлена и обсуждена на ряде научно-практических конференций [26–30].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Новости Worldsteel Association [сайт]. Режим доступа: <https://www.worldsteel.org/media-centre/press-releases/2016/--World-crude-steel-output-decreases-by--2.8--in-2015.html> .
2. Компьютерные методы моделирования доменного процесса / О.П. Онорин, Н.А. Спирин, В.Л. Терентьев [и др.]; под ред. Н.А. Спирина. – Екатеринбург: УГТУ–УПИ, 2005. – 301 с.
3. Цирлин А.М. Оптимальное управление технологическими процессами. М.: Энергоатомиздат, 1986. – 400 с.
4. Оптимизация в технике / Г.Рейклеитис, А.Рейвиндран, К.Рэгсдел. – М.: Мир, 1986. – 350 с.
5. Ашманов С.А. Линейное программирование. – М.: Радио и связь, 1981. – 340 с.
6. Модельные системы поддержки принятия решений в АСУ ТП доменной плавки / Н.А. Спирин, В.В. Лавров, В.Ю. Рыболовлев [и др.]; под ред. Н.А. Спирина. – Екатеринбург: УрФУ, 2011. – 456 с.
7. Оптимизация и идентификация технологических процессов в металлургии / Н.А. Спирин, В.В. Лавров, С.И. Паршаков [и др.]; под ред. Н.А. Спирина. – Екатеринбург: УГТУ–УПИ, 2006. – 307 с.
8. Информационные системы в металлургии / Н.А. Спирин, В.В. Лавров. – Екатеринбург: УПИ, 2004. – 495 с.
9. Кетков Ю.Л. MATLAB 7: программирование, численные методы. – СПб.: БХВ-Петербург, 2005. – 752 с.
10. Бахтизин В.В. Технология разработки программного обеспечения: учеб. пособие / В.В. Бахтизин, Л.А. Глухова. – Минск: БГУИР, 2010. – 267 с.
11. Свободные инструменты проектирования информационных систем / А.И. Попов; Сев. (Арктич.) федер. ун-т. им. М.В. Ломоносова – Архангельск: ИПЦ САФУ, 2012. 78с.

- 12.Одинцов И.О. Профессиональное программирование. Системный подход. 2-е изд. перераб. и доп. / И.О. Одинцов. – СПб.: БХВ-Петербург, 2004. – 624 с.
- 13.Методология функционального моделирования IDEF0. Госстандарт России. – М.: ИПК «Издательство стандартов», 2000. – 75 с.
<http://www.nsu.ru/smk/files/idef.pdf>
- 14.Горлушкина Н.Н. Системный анализ и моделирование информационных процессов и систем. Учебное пособие. – СПб: Университет ИТМО, 2016. – 120 с.
- 15.Макконнелл С. Совершенный код. Мастер-класс [Текст]: [пер. с англ.] / С.Макконнелл. – СПб.: Питер, 2007. – 896 с.
- 16.Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем: учебное пособие. – СПб.: Университет ИТМО, 2015. – 206 с.
- 17.Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. – Проектирование информационных систем: учебное пособие / 2-е изд., испр. – М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2010. – 299 с.
- 18.Маклаков С.В., Туманов В.Е. Проектирование реляционных хранилищ данных. – М.: Диалог–МИФИ, 2007. – 336 с.
- 19.Советов Б.Я. Базы данных: теория и практика: учебник для бакалавров / 2-е изд. – М.: Юрайт, 2012. – 464 с.
- 20.Ицик Бен-Ган. Microsoft SQL Server 2008. Основы T-SQL. – СПб.: БХВ-Петербург, 2009. – 430 с.
- 21.ИНТУИТ Национальный институт. Курс лекций Основы SQL
<http://www.intuit.ru/studies/courses/5/5/lecture/144?page=1>.
- 22.Pro Git. Шакон С. (Chacon Scott). URL: <https://git-scm.com/book/ru/v1>
- 23.Agile! Прекрасный, ужасный, шумный / Бертран Мейер; пер. с англ. и ред. В. Биллига. – Москва, Тверь: НАУ «ИНТУИТ». ЗАО НИИ ЦПС, 2015. – 248 с.
- 24.Гамильтон Б. ADO.NET. Сборник рецептов. – СПб.: Питер, 2005. – 576 с.

25. Балена Ф. Современная практика программирования на Microsoft Visual Basic и Visual C# [Текст]: [пер. с англ.] / Ф.Балена, Дж.Димауро. М.: Русская редакция, 2006. – 640 с.
26. Шпаргалка по ZedGraph [сайт]. URL: <http://jenyay.net/Programming/ZedGraph>. Дата обращения: 10.03.2016 г.
27. Решение задачи оптимизации в среде программирования Microsoft Visual Studio (C#) / Гурин И.А., Спирин Н.А., Лавров В.В., Бякова М.В. Системы автоматизации в образовании, науке и производстве: Труды X Всероссийской научно-практической конференции (17-19 декабря 2015 г.). Сиб. гос. индустр. ун-т. – Новокузнецк: изд. центр СибГИУ, 2015. С. 41–45.
28. Решение задачи оптимизации в среде программирования Microsoft Visual Studio (C#) / Гурин И.А., Спирин Н.А., Лавров В.В., Бякова М.В. Системы автоматизации в образовании, науке и производстве: Труды X Всероссийской научно-практической конференции (17-19 декабря 2015 г.). Сиб. гос. индустр. ун-т. – Новокузнецк: изд. центр СибГИУ, 2015. С. 41–45.
29. Программное обеспечение в АСУП доменного цеха / В.В. Лавров, Н.А. Спирин, А.А. Бурыкин, К.А. Щипанов, В.Ю. Рыболовлев, А.В. Краснобаев, М.А. Бякова. Современные научные достижения металлургической теплотехники и их реализация в промышленности: сборник докладов Международной научно-практической конференции, посвященной 95-летию основания кафедры ТИМ, УрФУ и 85-летию основания ОАО «ВНИИМТ» (Екатеринбург, 17–18 сентября 2015 г.). – Екатеринбург: УрФУ, 2015. С. 144–152.
30. Решение оптимизационных задач на языке программирования Visual C# с использованием математических пакетов / Гурин И.А., Спирин Н.А., Лавров В.В., Бякова М.В. Моделирование и наукоемкие информационные технологии в технических и социально-экономических системах: труды IV Всероссийской научно-практической конференции (12–15 апреля 2016 г.). В 2 ч. Ч1 / Сиб. гос. индустр. ун-т.; под общей редакцией В.П. Цымбала, Т.В. Киселевой. – Новокузнецк: изд. центр СибГИУ, 2016. С. 70–74.

- 31.Технология и программный инструментарий для разработки информационно-моделирующих систем доменного производства ОАО «ММК» / Лавров В.В., Спирин Н.А., Бурыкин А.А., Рыболовлев В.Ю., Истомин А.С., Гурин И.А., Бякова М.В. Моделирование и наукоемкие информационные технологии в технических и социально-экономических системах: труды IV Всероссийской научно-практической конференции (12–15 апреля 2016 г.). В 2 ч. Ч1 / Сиб. гос. индустр. ун-т.; под общей редакцией В.П. Цымбала, Т.В. Киселевой. – Новокузнецк: изд. центр СибГИУ, 2016. С. 179–186.
- 32.Разработка программного обеспечения для расчета оптимального распределения природного газа в группе доменных печей / М.А. Бякова, И.А. Гурин, В.В. Лавров, Н.А. Спирин. Теплотехника и информатика в образовании, науке и производстве: сборник докладов V Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (ТИМ'2016) с международным участием (Екатеринбург, 12–13 мая 2016 г.). – Екатеринбург: УрФУ, 2016. С. 179–183.
- 33.Лошкарев Н.Б. Указания к оформлению дипломных и курсовых проектов и работ. Методические указания / Н.Б. Лошкарев, А.Н. Лошкарев, Л.А. Зайнуллин. – Екатеринбург: УГТУ–УПИ, 2007. – 49 с.

ПРИЛОЖЕНИЕ А

Фрагменты листинга программного кода

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

namespace OptimaGaz
{
    public partial class FormMain : Form
    {
        public List<t_sprav_indicators> sprav_indicators;
        public List<t_data_indicators> data_indicators;

        public Dictionary<int, string, double> nid;

        Dictionary<int, string, double> limit_min = new Dictionary<int, string, double>();
        Dictionary<int, string, double> limit_max = new Dictionary<int, string, double>();

        public List<int> pech_to_checkedPech = new List<int>();

        public SqlCommand cmd;

        public FormMain()
        {
            InitializeComponent();

            // Get pechi
            SqlCommand cmd = new SqlCommand("SELECT pech_id, name FROM
t_sprav_pechi WHERE 1 = 1 ORDER BY name ASC", Program.conn);

            try
            {
                using (SqlDataReader reader = cmd.ExecuteReader())
                {
                    while( reader.Read() )
                    {
                        Program.pechi[ Int32.Parse(reader["pech_id"].ToString())
] = reader["name"].ToString();
                    }
                }
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

```

        MessageBox.Show("Ошибка: " + ex.Message);
    }

    // Заполняем ячейки печей
    foreach (KeyValuePair<int, string> curr in Program.pechi)
    {
        checkedList_Pechi.Items.Add(curr.Value, true);
        pech_to_checkedPech.Add(curr.Key);

        // test
        cmbManual.Items.Add(curr.Value);
        cmbManual.CheckBoxItems[cmbManual.Items.Count - 1].Checked =
true;
    }

    cmbManual.CheckBoxCheckedChanged += new
System.EventHandler(this.cmbManual_CheckedChanged);
}

// YK: Start
private void cmbManual_CheckedChanged(object sender, EventArgs e)
{
    CheckBox box = (CheckBox)sender; // get the checkbox affected

    // Brute force search in the datagrid columns
    foreach (DataGridViewColumn column in dataGrid_Indicators.Columns)
    {
        // We want to find the column with header text indicated in the
affected checkbox
        if (column.HeaderText == box.Text)
        {
            column.Visible = (box.CheckState == CheckState.Checked);
            break;
        }
    }
}
// YK: end

private void FormMain_Load(object sender, EventArgs e)
{
    saveToExcel();
    foreach (KeyValuePair<int, string> curr in Program.pechi)
    {
        DataGridViewTextBoxColumn col = new DataGridViewTextBoxColumn();
        col.Name = "pech" + curr.Key.ToString();
        col.HeaderText = curr.Value;
        col.Width = 77;

        dataGrid_Indicators.Columns.Add(col);
    }

    // Get indicators
    sprav_indicators = dataLib.get_sprav_indicaors();

    // Get data
    data_indicators = dataLib.get_data_indicators();
}

```

```

#region -- Запись в эксель ограничений на ДП
public void saveToExcel()
{
    limit_min = dataLib.get_arr_data_pechi_limits("min");
    limit_max = dataLib.get_arr_data_pechi_limits("max");

    int i = 0;
    // открыть эксель
    string folder =
Path.Combine(Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location),
    "excel5.xlsm");

    Excel.Application oExcel = new Excel.Application();
    var workbooks = oExcel.Workbooks;
    var WorkBook = workbooks.Open(folder);
    var WorkSheet2 = WorkBook.Worksheets["1"];

    // Содержание Серы
    foreach (KeyValuePair<int, string> curr in Program.pechi)
    {
        int currCount = i + 2;
        ((Microsoft.Office.Interop.Excel.Range)WorkSheet2.Cells[24,
currCount]).Value2 = limit_min[curr.Key, "SERA"].ToString();
        ((Microsoft.Office.Interop.Excel.Range)WorkSheet2.Cells[25,
currCount]).Value2 = limit_max[curr.Key, "SERA"].ToString();
        i++;
    }
    i = 0;
    // Содержание кремния
    foreach (KeyValuePair<int, string> curr in Program.pechi)
    {
        int currCount = i + 2;
        ((Microsoft.Office.Interop.Excel.Range)WorkSheet2.Cells[22,
currCount]).Value2 = limit_min[curr.Key, "SI"].ToString();
        ((Microsoft.Office.Interop.Excel.Range)WorkSheet2.Cells[23,
currCount]).Value2 = limit_max[curr.Key, "SI"].ToString();
        i++;
    }
try
    {
        WorkBook.Save();
        WorkBook.Close();
        oExcel.Quit();
        Marshal.ReleaseComObject(WorkBook);
        Marshal.ReleaseComObject(workbooks);
        Marshal.ReleaseComObject(oExcel);
        oExcel = null;
        workbooks = null;
        WorkBook = null;
        GC.Collect();
    }
    catch (Exception ex)
    {
        WorkBook.Close();
        oExcel.Quit();
    }
}

```

```

        Marshal.ReleaseComObject(WorkBook);
        Marshal.ReleaseComObject(workbooks);
        Marshal.ReleaseComObject(oExcel);
        oExcel = null;
        workbooks = null;
        WorkBook = null;
        GC.Collect();
    }
}
#endregion -- Запись в эксель ограничений на ДП
// -- обновление dataGridView из объекта data_indicators
private void update_dgv_data_indicators()
{
    dataGrid_Indicators.Rows.Clear();

    Dictionary<int, int> indicator_to_row = new Dictionary<int, int>();

    sprav_indicators.ForEach(delegate (t_sprav_indicators data)
    {
        int index = dataGrid_Indicators.Rows.Add();
        indicator_to_row[data.indicator_id] = index;

        dataGrid_Indicators.Rows[index].Cells["name"].Value =
data.name.ToString();
        dataGrid_Indicators.Rows[index].Cells["indicator_id"].Value =
data.indicator_id.ToString();
    });

    data_indicators.ForEach(delegate (t_data_indicators data)
    {
        int rowId = indicator_to_row[data.indicator_id];

        dataGrid_Indicators.Rows[rowId].Cells["pech" +
data.pech_id.ToString()].Value = String.Format("{0:0.#####}", data.value);
        //data.value.ToString("0.#####",
System.Globalization.CultureInfo.InvariantCulture); //ToString("F5");
    });
}
private void button_test_save_Click(object sender, EventArgs e)
{
    try
    {
        cmd = new SqlCommand("TRUNCATE TABLE t_data_indicators",
Program.conn);
        cmd.ExecuteNonQuery();

        for (int i = 0; i < dataGrid_Indicators.Rows.Count; i++)
        {
            foreach (KeyValuePair<int, string> curr in Program.pechi)
            {
                cmd = new SqlCommand("INSERT INTO t_data_indicators
(pech_id, indicator_id, value) VALUES (@pech_id, @indicator_id, @value)",
Program.conn);

                cmd.Parameters.Add(new SqlParameter { SqlDbType =
SqlDbType.Int, ParameterName = "@pech_id", Value = curr.Key });
            }
        }
    }
}

```

```

        cmd.Parameters.Add(new SqlParameter { SqlDbType =
SqlDbType.Int, ParameterName = "@indicator_id", Value =
dataGrid_Indicators.Rows[i].Cells["indicator_id"].Value });
        cmd.Parameters.Add(new SqlParameter { SqlDbType =
SqlDbType.Decimal, ParameterName = "@value", Value =
dataGrid_Indicators.Rows[i].Cells["pech" + curr.Key.ToString()].Value });

        cmd.ExecuteNonQuery();
    }
}

MessageBox.Show("Сохранено...");
}
catch (Exception ex)
{
    MessageBox.Show("Ошибка при сохранении данных в БД (" +
ex.Message + ")");
}
}

```

```

private void checkedList_Pechi_SelectedIndexChanged(object sender,
EventArgs e)
{
    for (int i = 0; i < checkedList_Pechi.Items.Count; i++)
    {
        string pech = "pech" + pech_to_checkedPech[i].ToString();

        if (checkedList_Pechi.GetItemCheckState(i) ==
CheckState.Checked) dataGrid_Indicators.Columns[pech].Visible = true;
        else dataGrid_Indicators.Columns[pech].Visible = false;
    }
}

```

```

//Запись в Excel
private void resultStripMenuItem_Click(object sender, EventArgs e)
{
    string folder =
Path.Combine(Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location),
        "excel5.xlsm");

    Excel.Application oExcel = new Excel.Application();
    var workbooks = oExcel.Workbooks;
    var WorkBook = workbooks.Open(folder);
    var WorkSheet = WorkBook.Worksheets["1"];
    //oExcel.Visible = false;

    for (int i = 2; i < dataGrid_Indicators.ColumnCount; i++)
    {
        for (int j = 0; j < dataGrid_Indicators.RowCount; j++)
        {
            ((Microsoft.Office.Interop.Excel.Range)WorkSheet.Cells[j +
4, i]).Value2 =
                dataGrid_Indicators.Rows[j].Cells[i].Value.ToString();
        }
    }
}

```



```

//Console.WriteLine(dataGrid_Indicators.Rows[j].Cells[i].Value.ToString());

    }
}
try //попытаться выполнить код, если не выполняется, то.....
{
    oExcel.Visible = false;
    oExcel.GetType().
        InvokeMember("Run",
            System.Reflection.BindingFlags.Default |
            System.Reflection.BindingFlags.InvokeMethod,
            null, oExcel, new object[] { "Raschet" });
}
catch (Exception ex) //....вызывается вот это
{
    MessageBox.Show("Ошибка (" + ex.Message + ")");
}

    double totalResult = Convert.ToDouble((WorkSheet.UsedRange.Cells[7,
11] as Microsoft.Office.Interop.Excel.Range).Value2);
    double OptimaResult =
Convert.ToDouble((WorkSheet.UsedRange.Cells[13, 11] as
Microsoft.Office.Interop.Excel.Range).Value2);
    double RezervResult = Convert.ToDouble((WorkSheet.UsedRange.Cells[6,
13] as Microsoft.Office.Interop.Excel.Range).Value2);
    double checkResult = Convert.ToDouble((WorkSheet.UsedRange.Cells[60,
5] as Microsoft.Office.Interop.Excel.Range).Value2);

    Dictionary<int, double> resultDict = new Dictionary<int, double>();
    for (int i = 0; i < 8; i++)
    {
        int cellNumber = i + 2;
        resultDict.Add(i,
Convert.ToDouble((WorkSheet.UsedRange.Cells[13, cellNumber] as
Microsoft.Office.Interop.Excel.Range).Value2));
    }
    resultDict.Add(8, totalResult);
    resultDict.Add(9, checkResult);
    resultDict.Add(10, OptimaResult);
    resultDict.Add(11, RezervResult);
if (checkResult < 4)
{
    ResultsForm resultForm = new ResultsForm(resultDict, baseDict,
        koksRashod, koksRashodBase,
        obemChug, obemChugBase,
        tempGor, tempGorBase,
        siChug, siChugBase,
        sChug, sChugBase,
        teploShiht, teploShihtBase,
        stepShiht, stepShihtBase);

    MessageBoxButtons buttons = MessageBoxButtons.OK;
    //string strMsgMessage = "Внимание!" + "\n" + "Ошибка при
подключении к источнику данных!";
    string strMsgMessage = "Решение найдено!";
}
}
}

```

```

        MessageBox.Show(strMsgMessage, "Сообщение", buttons,
        MessageBoxIcon.Information);
        //MessageBox.Show("Решение найдено");

        resultForm.Show();
    }
    else
    {
        MessageBox.Show("Решение не найдено");
    }

    try
    {
        Workbook.Save();
        Workbook.Close();
        oExcel.Quit();
        Marshal.ReleaseComObject(Workbook);
        Marshal.ReleaseComObject(workbooks);
        Marshal.ReleaseComObject(oExcel);
        oExcel = null;
        workbooks = null;
        Workbook = null;
        GC.Collect();
    }
    catch (Exception ex)
    {
        Workbook.Close();
        oExcel.Quit();
        Marshal.ReleaseComObject(Workbook);
        Marshal.ReleaseComObject(workbooks);
        Marshal.ReleaseComObject(oExcel);
        oExcel = null;
        workbooks = null;
        Workbook = null;
        GC.Collect();
    }
}

private void RunMacro(object oApp, object[] oRunArgs)
{
    oApp.GetType().InvokeMember("Raschet",
        System.Reflection.BindingFlags.Default |
        System.Reflection.BindingFlags.InvokeMethod,
        null, oApp, oRunArgs);
}

```