

ИНСТРУКЦИЯ ПО РАБОТЕ С ПРОГРАММОЙ

Алгоритм поиска приближенного решения задачи размещения взаимосвязанных габаритных объектов на линии с запрещенными зонами

СОДЕРЖАНИЕ

Введение.....	2
1. Постановка задачи.....	2
2. Описание алгоритма.....	4
2.1 Поиск допустимых разбиений.....	4
2.2 Минимизация суммарной стоимости связей.....	5
3. Описание работы с программой.....	5
3.1 Восстановление базы данных из backup.....	7
3.2 Установление связи программы с базой данных.....	15
3.3 Работа пользователя с ПО.....	16
3.3.1 Справочник задач для поиска приближенного решения на линии.....	17
3.3.2 Решение алгоритмом поиска приближенного решения.....	19

Введение

Вашему вниманию предлагается инструкция по работе с программой поиска приближенного решения задачи размещения взаимосвязанных габаритных объектов на линии с запрещенными зонами. Актуальными в настоящее время являются задачи оптимального размещения объектов в заданной области. Такие задачи необходимо решать при проектировании генеральных планов предприятий, расстановке технологического оборудования в цехах каких-либо производств, расположении пунктов обслуживания и т.д. Часто, для создания прямых проездов, удобства обслуживания оборудования, требуется условие регулярности размещения, например, вдоль заданных осевых линий. Кроме того, могут быть участки (запрещенные зоны), в которых нельзя размещать оборудование по каким-либо причинам, например, при модернизации предприятия часть оборудования, которая остается на месте. Рассматриваемая задача может применяться в автоматизации и проектировании технологических линий при реконструкции или замене оборудования в цехе предприятия.

1. Постановка задачи

Опишем постановку задачи.

Заданы: отрезок длины LS с фиксированными на нём запрещенными зонами F_1, \dots, F_m с координатами центров b_j и длинами $p_j, j \in J = \{1, \dots, m\}$;

набор размещаемых объектов X_1, \dots, X_n с неизвестными координатами центров x_i и длинами $l_i, i \in I = \{1, \dots, n\}$;

$w_{ij} \geq 0, u_{ik} \geq 0$ – удельные стоимости связей между $X_i, F_j, i \in I, j \in J$, и $X_i, X_k, i, k \in I, i < k$, соответственно.

Необходимо разместить объекты X_1, \dots, X_n на отрезке вне зон F_1, \dots, F_m так, чтобы они не пересекались, и суммарная стоимость связей объектов между собой и с зонами была минимальной. Математическая модель имеет следующий вид.

$$G(x) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} |x_i - b_j| + \sum_{i=1}^{n-1} \sum_{k=i+1}^n u_{ik} |x_i - x_k| \rightarrow \min, \quad (1)$$

$$|x_i - b_j| \geq \frac{l_i + p_j}{2}, \quad i \in I, j \in J, \quad (2)$$

$$|x_i - x_k| \geq \frac{l_i + l_k}{2}, \quad i, k \in I, i < k, \quad (3)$$

$$\frac{l_i}{2} \leq x_i \leq LS - \frac{l_i}{2}, \quad i \in I. \quad (4)$$

Первая составляющая в (1) определяет суммарную стоимость связей между объектами и зонами, вторая – между объектами, а (2) и (3) – это условия на непересечение объектов с зонами и между собой, (4) – условия расположения объектов на отрезке.

Допустимая область B несвязная и состоит из набора r непересекающихся отрезков (блоков) B_k с длинами L_k , в которые размещаются объекты X_i , $i \in I$, $B = \bigcup_{k=1, \dots, r} B_k$.

Задача (1)-(4) NP-трудная, поиск её допустимого решения – это построение одномерной упаковки. В данном случае упаковываются объекты с длинами l_i , $i \in I$, в контейнеры размерами L_k , $k = \overline{1, r}$.

Для допустимого решения будем называть *остатком* в блоке B_k отрезок между двумя соседними объектами без общей границы, либо между границей блока и соседним объектом. Блок без объектов будем считать блоком с остатком. Пары элементов (объекты, зоны, остатки) будем называть *склеенными*, если они имеют общую границу.

Обозначим $J_L(B_k)$, $J_R(B_k)$ – множество зон левее, правее блока B_k , $I_L(A)$, $I_R(A)$ – множество объектов левее, правее блока (зоны) A . Для каждого объекта X_i в блоке B_k определим суммарные стоимости связей Lw_i и Rw_i следующим образом

$$Lw_i = \sum_{j \in J_L(B_k)} w_{ij} + \sum_{k \in I_L(B_k)} u_{ik}, \quad Rw_i = \sum_{j \in J_R(B_k)} w_{ij} + \sum_{k \in I_R(B_k)} u_{ik}.$$

Пусть $x = (x_1, \dots, x_n)$ – некоторое допустимое решение задачи (1)-(4), которое однозначно определяет разбиение объектов X_1, \dots, X_n по блокам. Обозначим $I_k(x)$ – множество номеров объектов в B_k , $I = \bigcup_{k=1, \dots, r} I_k(x)$, а через $H_k(x)$ – совокупность остатков в B_k с номерами из $I_k(x)$. Пусть величина n_k задает мощность множества $I_k(x)$, $|I_k(x)| = n_k$, тогда $|H_k(x)| \leq n_k + 1$. Отметим, что x можно представить в виде $x = (x^1, \dots, x^r)$, где x^k – координаты центров объектов, расположенных в B_k с номерами из $I_k(x)$.

Утверждение 1. Для произвольного допустимого решения x задачи (1)-(4) при фиксированном разбиении объектов по блокам можно построить другое допустимое решение x' такое, что $|H_k(x')| \leq 1$, $k = \overline{1, r}$ и $G(x') \leq G(x)$.

Из утверждения 1 следует, что для заданного разбиения объектов по блокам и фиксированного порядка их расположения можно построить размещение с наименьшим значением целевой функции, имеющее не более одного остатка в блоке. Длину остатка в блоке B_k обозначим как Δ_k . Остаток в блоке можно рассматривать как дополнительный размещаемый объект X_{n_k+1} , для которого $l_{k+1} = \Delta_k$ и $Lw_{n_k+1} = Rw_{n_k+1} = 0$.

Обозначим LB_k , RB_k – координаты левой и правой границ блока B_k . Тогда при фиксированном разбиении объектов по блокам целевую функцию $G(x)$ можно представить в виде

Целевую функцию задачи можно представить в виде

$$G(x) = \sum_{k=1}^r G_k(x^k) + C, \text{ где}$$

$$G_k(x^k) = \sum_{s,t \in I_k, s < t} u_{st} \cdot |x_s - x_t| + \sum_{j \in J_L(B_k), s \in I_k} w_{sj} \cdot |x_s - L(B_k)| + \sum_{j \in J_R(B_k), t \in I_k} w_{jt} \cdot |x_t - R(B_k)|,$$

C – некоторая константа.

Первая составляющая в $G_k(x^k)$ – это суммарная стоимость связей между объектами в блоке B_k , вторая – между объектами из B_k и LB_k , третья – между объектами из B_k и RB_k .

Без ограничения общности далее всегда рассматривается случай, когда область размещения на отрезке ограничена слева и справа зонами.

Таким образом, для нахождения решения задачи (1)-(4) при фиксированном разбиении объектов по блокам, достаточно найти минимумы функций $G_k(x^k)$, $k = 1, \dots, r$.

2. Описание алгоритма

Итерация алгоритма состоит из двух этапов. На первом этапе находится очередное допустимое разбиение объектов по блокам с помощью алгоритма последовательно-одиночного размещения, на втором – объекты переставляются в блоках с целью минимизации суммарной стоимости связей. Критерии остановки алгоритма: время работы, просмотр всех возможных разбиений. Трудоемкость предложенного алгоритма можно оценить как $O(r^n(mn^2 + n^3))$.

2.1 Поиск допустимых разбиений

Блок B_k будем называть *допустимым* для объекта X_i , если $L_k \geq l_i$ и *условно-допустимым*, если $L_k \geq \sum_h l_h + l_i$, где $\sum_h l_h$ – суммарная длина объектов, размещенных в B_k . Блок B_k будем называть *просмотренным* для X_i , если при фиксированном размещении X_1, \dots, X_{i-1} , объект X_i размещался в B_k , иначе – *непросмотренным*.

Пусть $L_1 \geq \dots \geq L_r$, $l_1 \geq \dots \geq l_n$. В ходе работы алгоритма поиска допустимых разбиений (назовем его А1) для построения начального допустимого разбиения в заданном порядке для каждого объекта находится первый условно-допустимый блок. Если такого блока нет, то отменяется принадлежность блоку для предыдущего объекта и т.д. После отмены, для каждого следующего объекта поиск условно-допустимого блока начинается с B_1 . Если допустимое разбиение не построено и при этом для X_1 просмотрены все допустимые блоки, то задача не имеет решения.

Для построения очередного допустимого разбиения, начиная с X_n , осуществляется поиск следующих условно-допустимых блоков для объектов.

Количество вариантов разбиений $X_i, i \in I$, по блокам B_1, \dots, B_r не превосходит r^n . Отметим, что с помощью алгоритма А1 будут просмотрены всевозможные допустимые разбиения объектов по блокам. Действительно, при нахождении для произвольного объекта X_i следующего условно-допустимого блока просмотр блоков начинается с B_1 для всех $X_j, j > i$.

2.2 Минимизация суммарной стоимости связей

Предлагается вариант минимизации суммарной стоимости связей объектов с учетом их длин (А2) в блоке. Объекты в блоке последовательно склеиваются в зависимости от суммарной стоимости связей с объектами и зонами, расположенными левее (правее) блока. Обозначим $N_L(B_k)$ – множество объектов, склеенных между собой, самый левый из которых склеен с левой границей блока B_k , а $N_R(B_k)$ – соответственно с правой, и пусть объекты в B_k имеют номера от 1 до n_k .

Алгоритм А2

Шаг 0. $T := I_k(x); N_L(B_k) := \theta; N_R(B_k) := \theta;$

Шаг 1. Если $(N_L(B_k) = \theta)$ and $(N_R(B_k) = \theta)$, то на Шаг 2,

Иначе

Если $(t \in N_L(B_k))$, то $\forall i \in T \quad Lw_i := Lw_i + Lw_t$, на Шаг 2,

Иначе $\forall i \in T \quad Rw_i := Rw_i + Rw_t$.

Шаг 2. Определяем $t: \max_{i \in T} \frac{|Lw_i - Rw_i|}{l_i} = \frac{|Lw_t - Rw_t|}{l_t}$.

Шаг 3. Если $(Lw_t \geq Rw_t)$, то $x_t := LB_k + \sum_{i \in N_L(B_k)} l_i + \frac{l_t}{2}; N_L(B_k) := N_L(B_k) \cup \{t\};$

Иначе $x_t := RB_k - \sum_{i \in N_R(B_k)} l_i - \frac{l_t}{2}; N_R(B_k) := N_R(B_k) \cup \{t\}.$

Шаг 4. $T := T \cup \{t\};$

Если $(T \neq \theta)$, то на Шаг 1,

Иначе СТОП (объекты размещены в B_k).

3. Описание работы с программой

Алгоритм поиска приближенного решения задачи размещения габаритных объектов на линии с запрещенными зонами реализован в среде разработки Borland C++ Builder 6.0.

Хранение первоначальных данных и результатов решений задач организовано в виде базы данных (БД), созданной на основе MS SQL Server 2000. Схема структуры базы представлена на рисунке 1.

Все данные тестируемых задач логически разделены и объединены в соответствующие таблицы, например, tVanBlock – таблица запрещенных зон,

tGoodBlock – разрешенных областей (блоков), tW и tV – стоимости связей размещаемых объектов с зонами и размещаемых объектов между собой. Результаты решения задачи фиксируются в таблицах tNewObject – координаты центров размещаемых объектов, tGetFunc – значение целевой функции, время счета, количество итераций. Таблица tTask играет роль справочника всех тестируемых задач, она связывает все перечисленные выше таблицы по ключу IDTask. Возможность работы с БД осуществляется с помощью клиентской программы, которая предоставляет пользователю возможности ввода, просмотра, редактирования информации о тестируемых задачах.

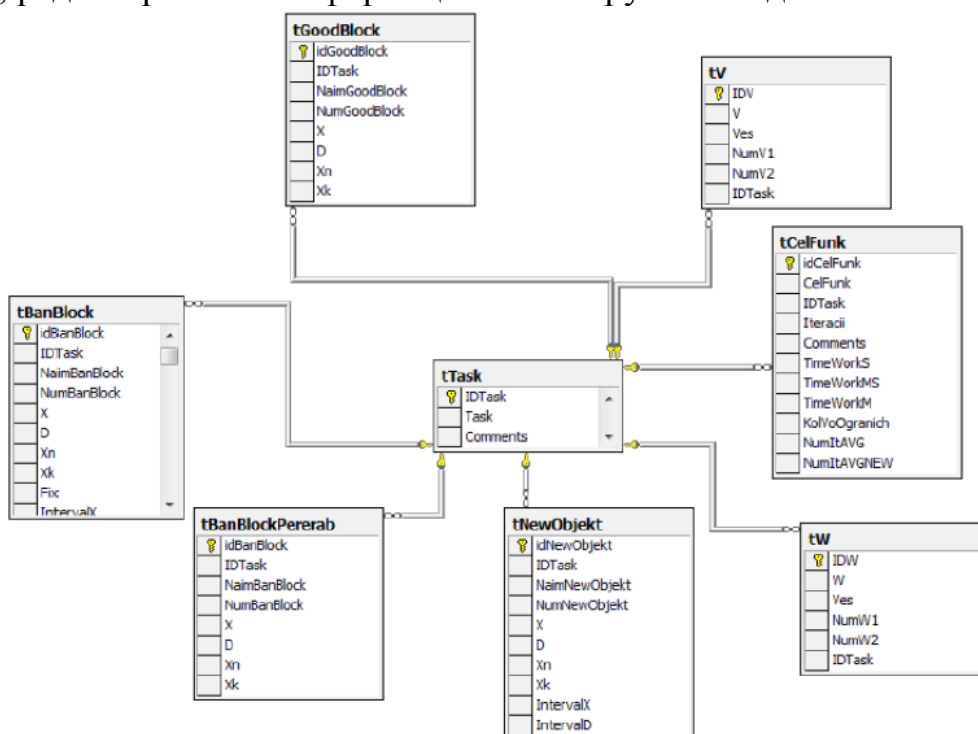


Рис.1. Схема структуры базы данных.

Требования к рабочей станции пользователя: операционная система Windows XP (и выше), минимальный размер оперативной памяти 4 ГБ, MS SQL Server 2000.

Необходимо, чтобы в папке с программой Road.exe находились все используемые в приложении шаблоны MS Excel (см. Рис. 2):

1. Block.xlt;
2. ForSimplex.xlt;
3. GoodBlock.xlt;
4. NewObjekt.xlt;
5. R.xlt;
6. Rez.xlt;
7. Statistics.xlt;
8. Statistics2.xlt;
9. Z.xlt.

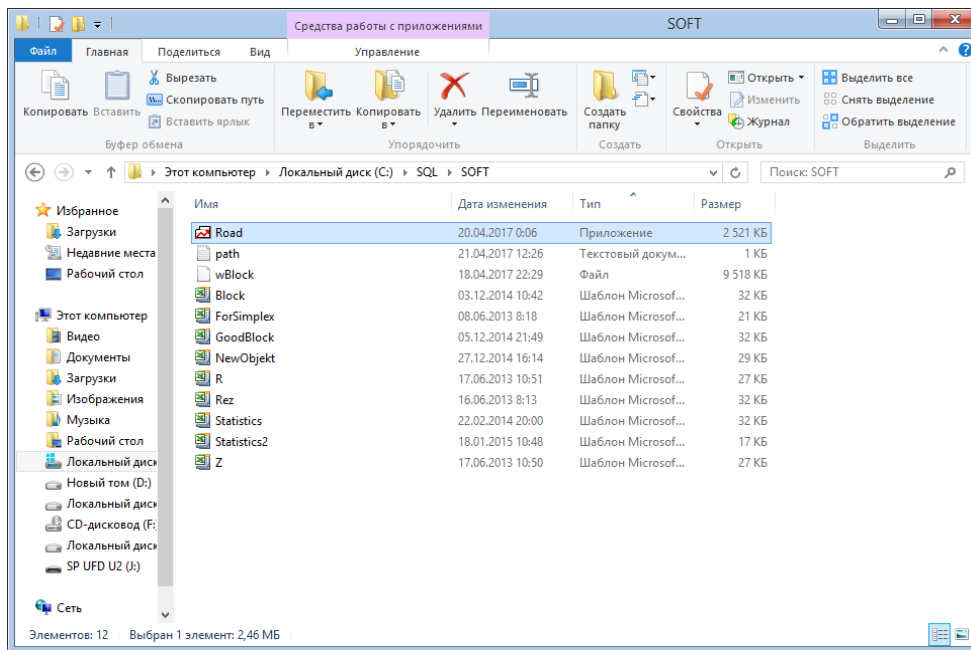


Рис. 2. Список файлов для работы приложения.

Для корректного запуска программы необходимо восстановить БД из backup wBlock и установить связь программного обеспечения (ПО) с БД. Предполагается, что MS SQL Server 2000 уже инсталлирован на компьютере.

3.1 Восстановление базы данных из backup

В меню Пуск, Программы находим Server Manager и запускаем его (см. Рис.3, Рис. 4).

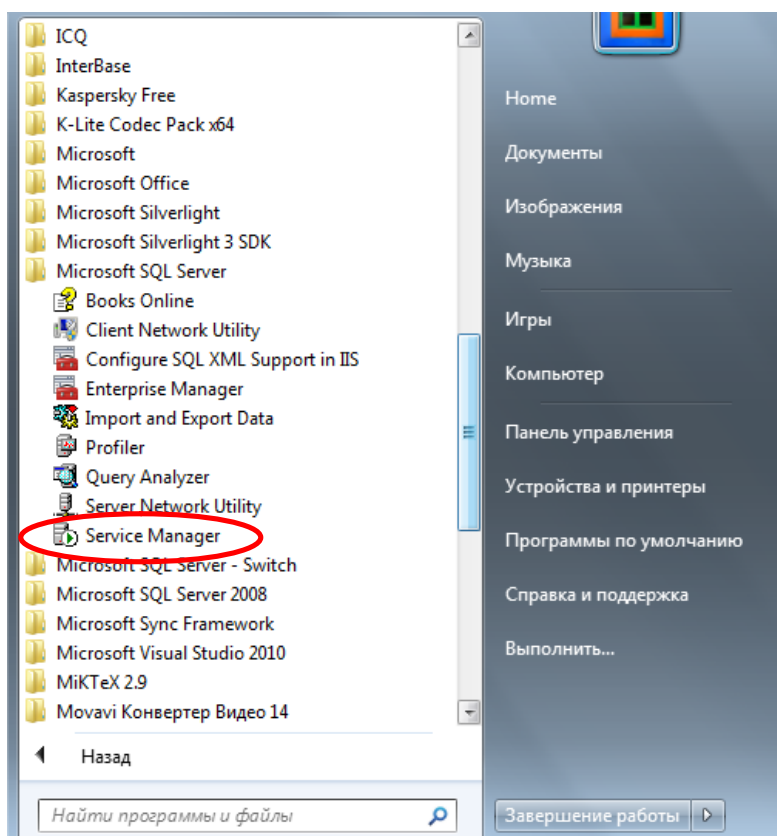


Рис. 3. Service Manager Для Windows 7.

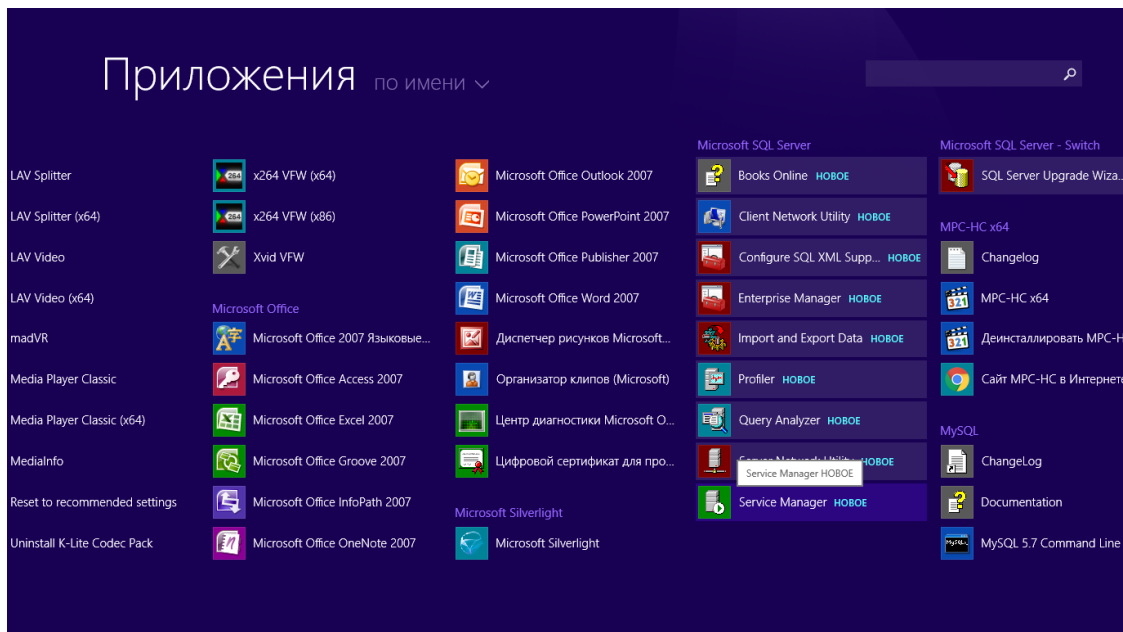


Рис. 4. Service Manager Для Windows 8.

Откроется следующая форма (см. Рис.5):

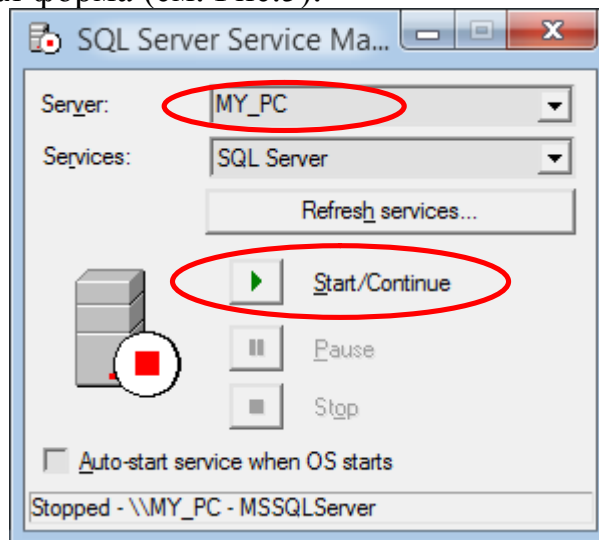


Рис. 5. SQL Server Service.

На форме видим, что наименование сервера по-умолчанию взялось автоматически по наименованию используемого ПК: MY_PC. Нажимаем кнопку *Start/Continue* - сервер запустится (см. Рис.6).

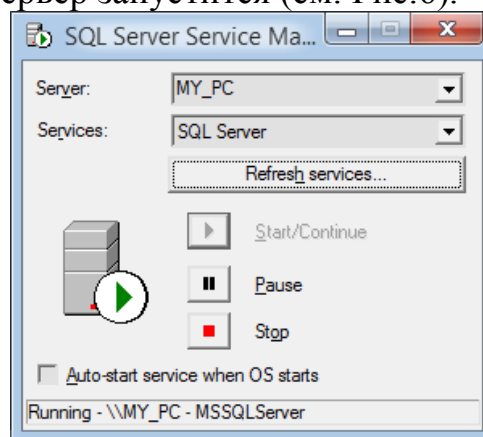


Рис. 6. Запуск сервера.

Ищем в программах *Enterprise Manager* и запускаем его (см. Рис. 7).

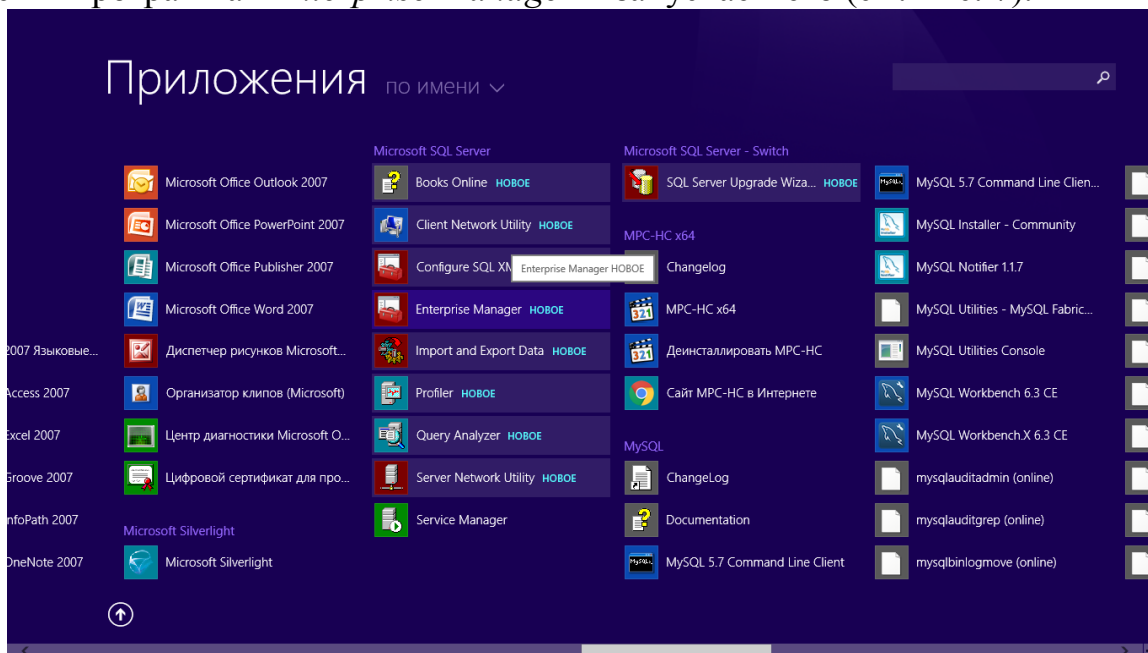


Рис. 7. Запуск *Enterprise Manager*.

Откроется *Enterprise Manager* (см. Рис. 8)

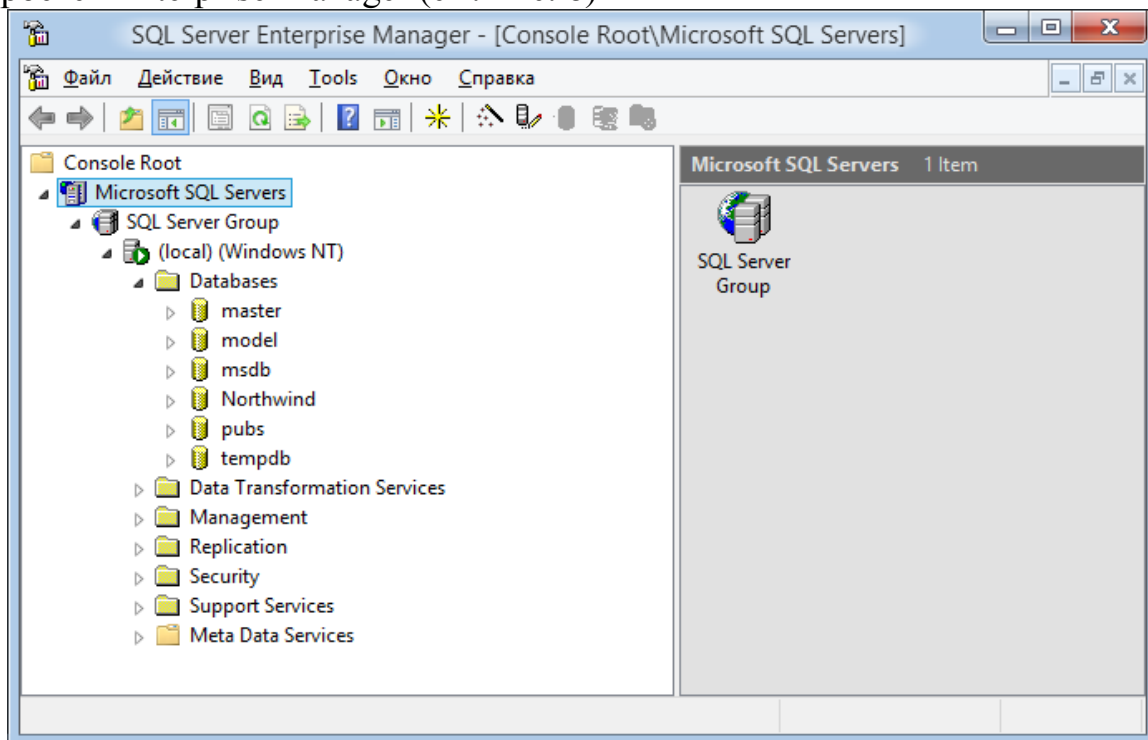


Рис. 8. Рабочая форма *Enterprise Manager*.

Создадим пустую БД с именем wBlock, для этого в папке Database нажимаем правую кнопку мыши и выбираем пункт NewDatabase (см. Рис. 9).

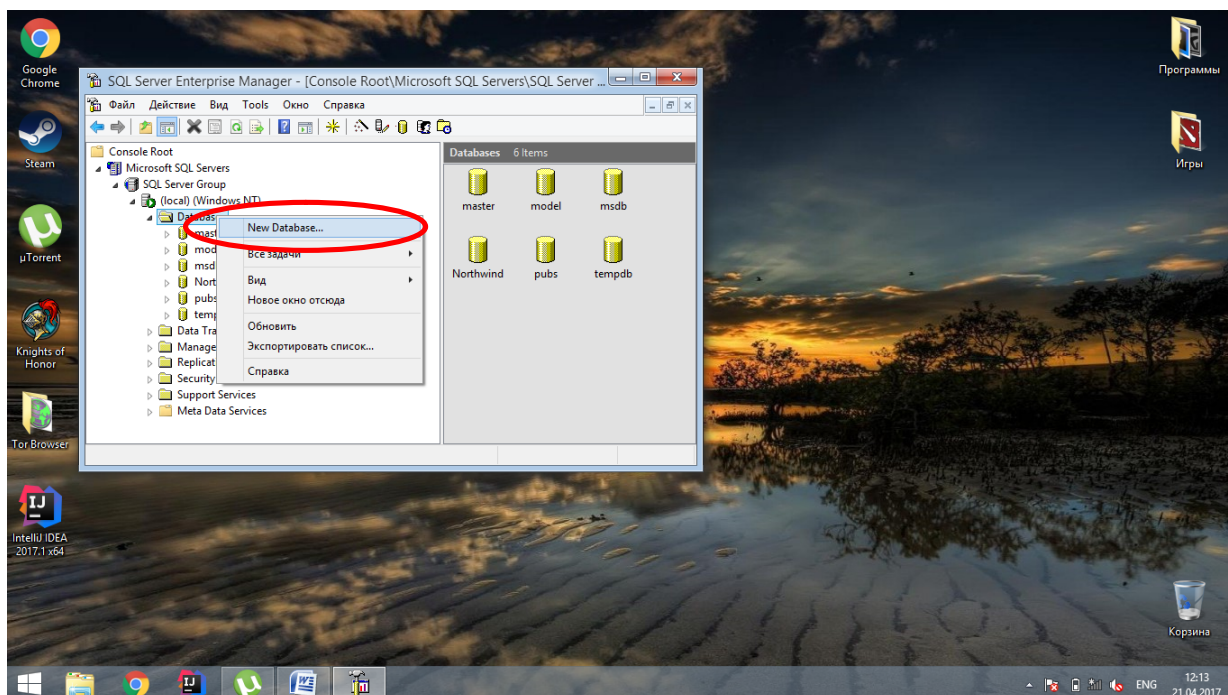


Рис. 9. Создание пустой базы данных в *Enterprise Manager*.

После этого возникает следующая форма (см. Рис. 10). В поле *Name* нужно ввести wBlock – это будет наименование БД и нажать ОК.

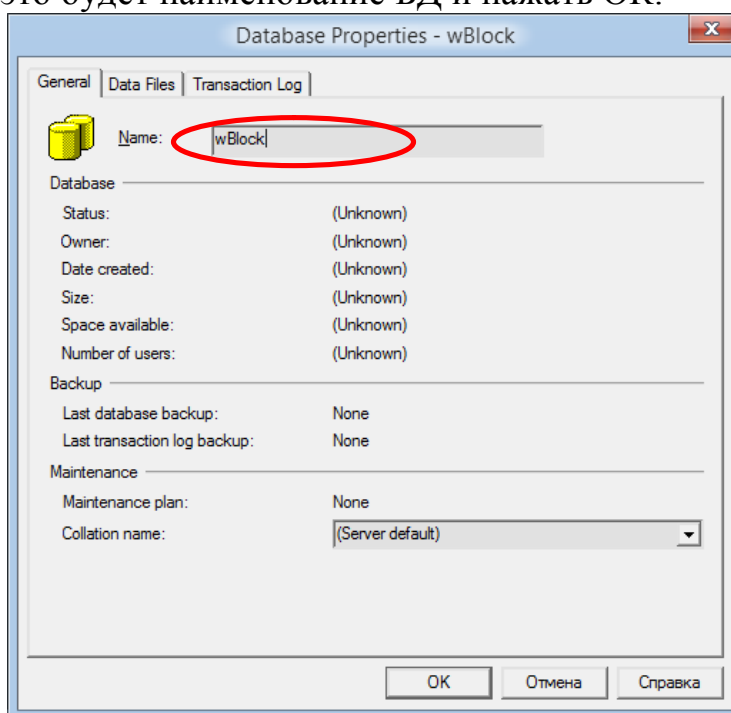


Рис. 10. Создание базы wBlock.

После этого на форме *Enterprise Manager* в дереве баз данных появиться наша база wBlock (см. Рис. 11).

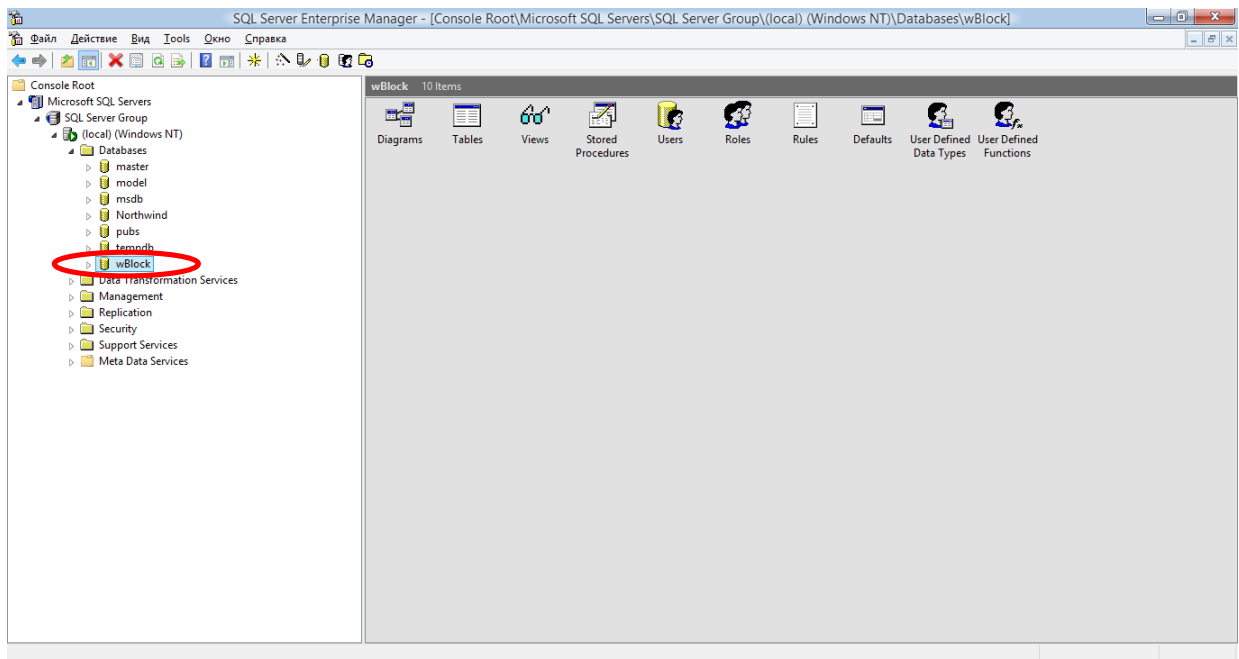


Рис. 11. Созданная база wBlock.

Далее, нужно восстановить БД wBlock из backup. Для этого в этом же дереве баз данных открытой формы Enterprise Manager выбираем только что созданную БД wBlock и нажимаем на нее правой клавишей мыши. В открывшемся контекстном меню выбираем пункт *Все задачи* → *Restore Database* (см. Рис. 12).

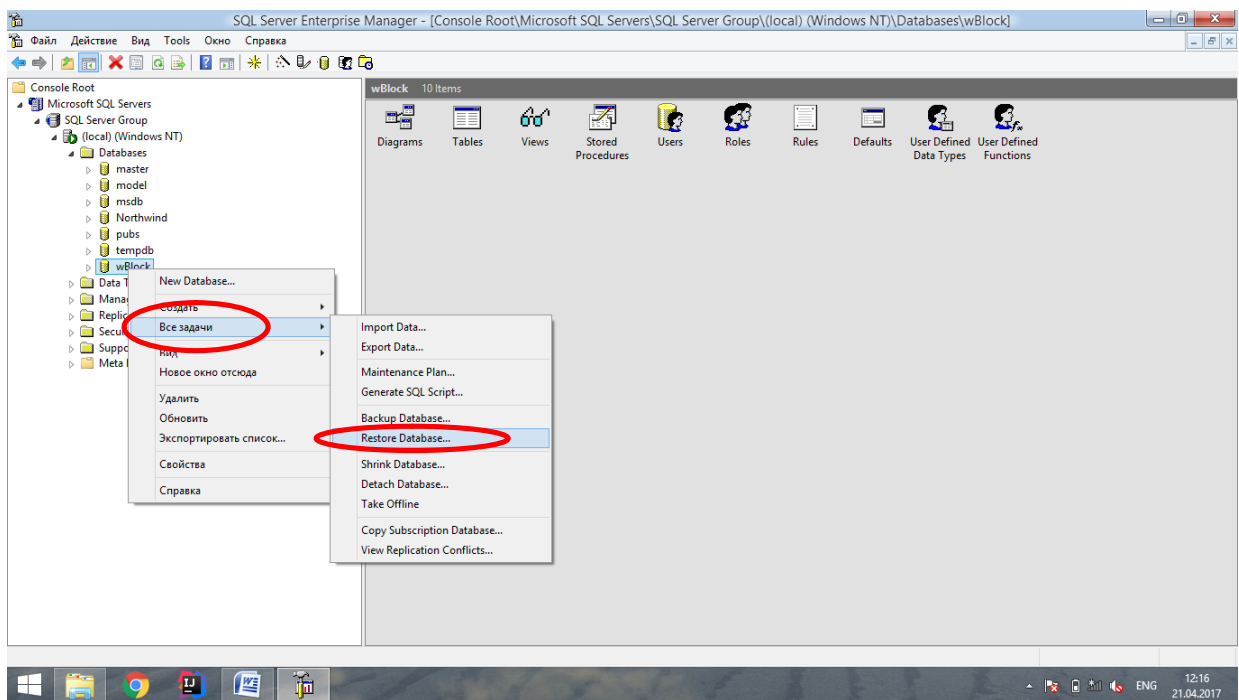


Рис. 12. Пункт *Restore Database*.

Откроется форма *Restore database* (см Рис. 13).

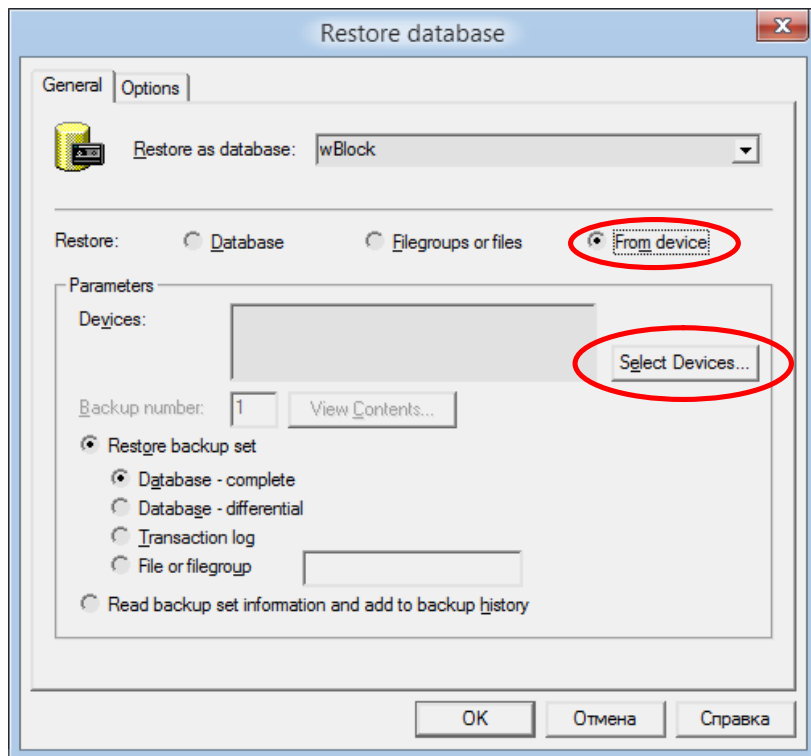


Рис. 13. Форма *Restore database*.

На форме *Restore database* ставим галочку *From device*. Нажимаем кнопку *Select Devices*. Откроется следующая форма *Choose Restore Devices*, где нажимаем кнопку *Add* (см. Рис. 14).

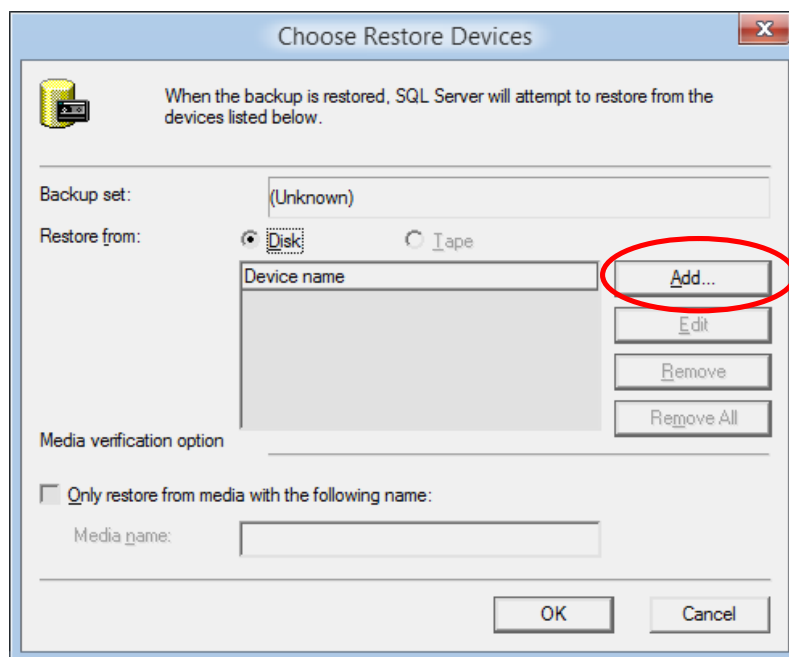


Рис. 14. *Choose Restore Devices*.

Откроется форма *Chose Restore Destination*, на которой ставим галочку в поле *File name*, нажимаем кнопку с "..." (см. Рис. 15).

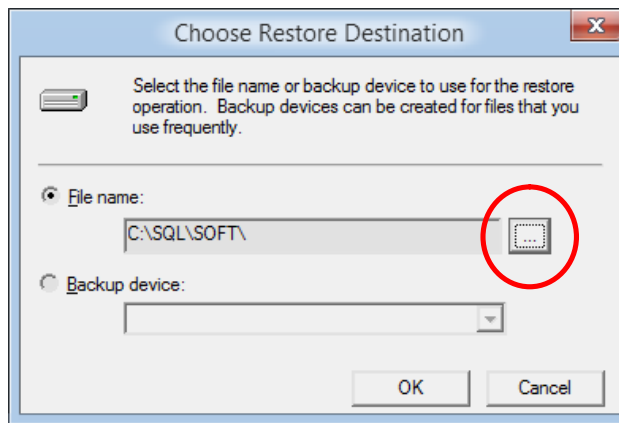


Рис. 15. Форма *Chose Restore Destination*.

Откроется форма *Backup Device Location* с диалоговым окном выбора файлов, где выбираем backup wBlock и нажимаем кнопку *OK* (см. Рис. 16).

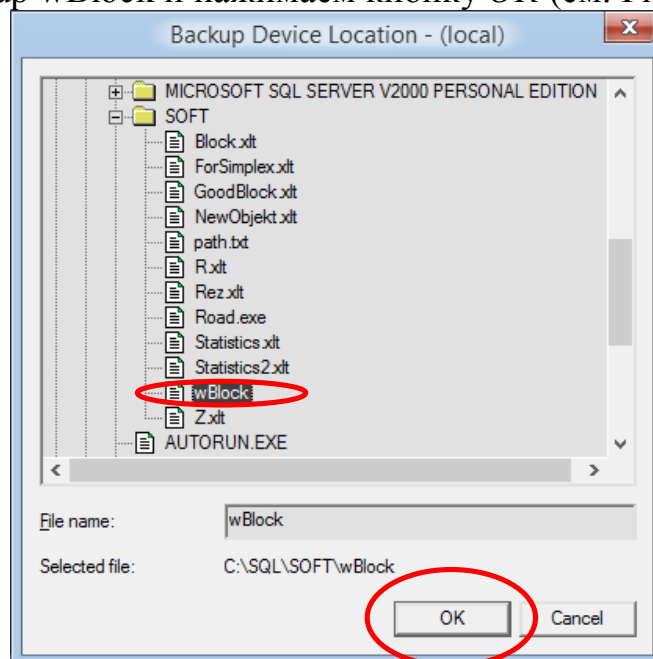


Рис. 16. Форма *Backup Device Location*.

Пошёл процесс восстановления БД из backup wBlock (см Рис. 17).

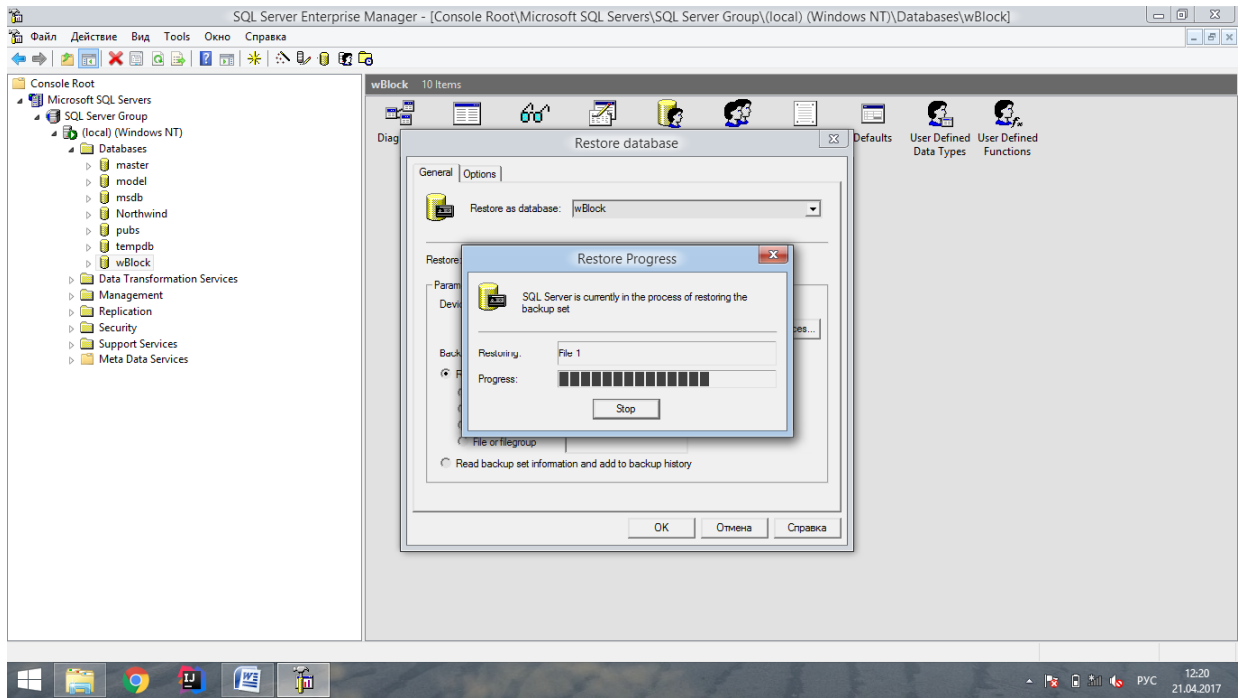


Рис. 17. Процесс восстановления БД из backup wBlock.

После успешного восстановления БД возникает информационное сообщение. Нажимаем *ОК* (см. Рис. 18).

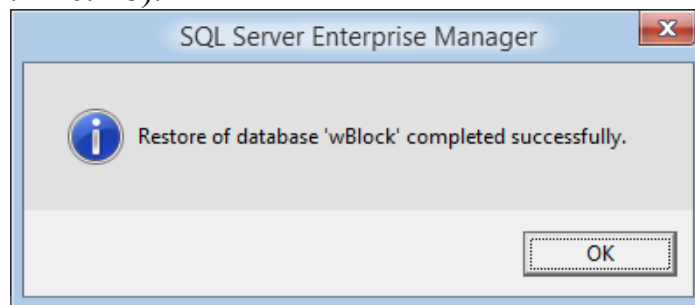


Рис. 18. Информационное сообщение.

Удостоверяемся, что база wBlock успешно установилась, открыв ее содержимое в *Enterprise Manager* (см. Рис. 19).

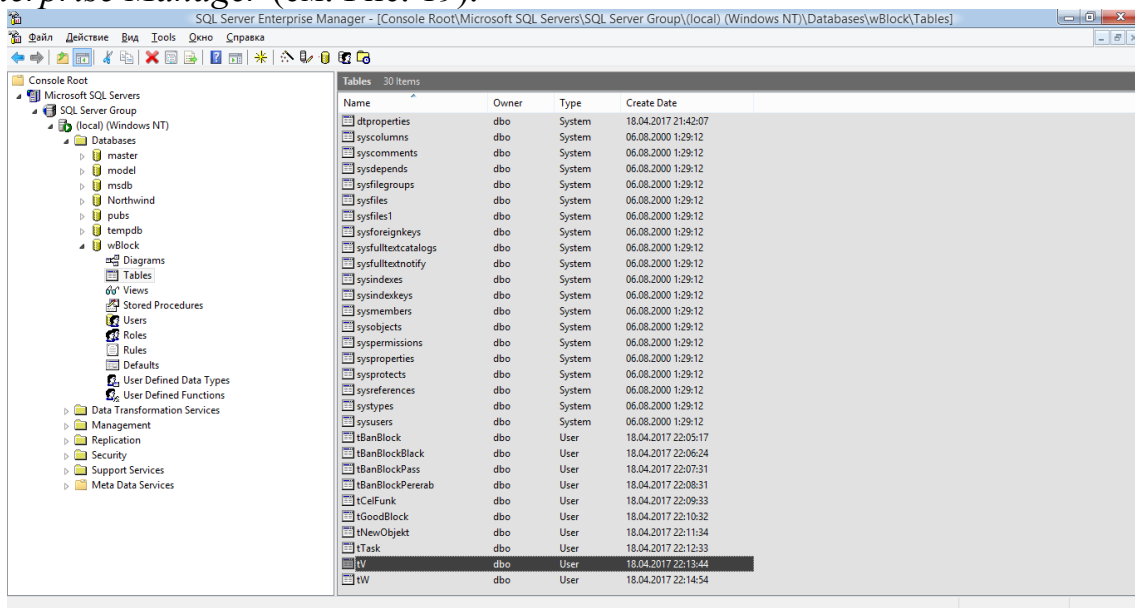


Рис. 19. Содержимое БД wBlock в *Enterprise Manager*.

3.2 Установление связи программы с базой данных

Для работы с программой запускаем приложение Road.exe (см. Рис. 20).

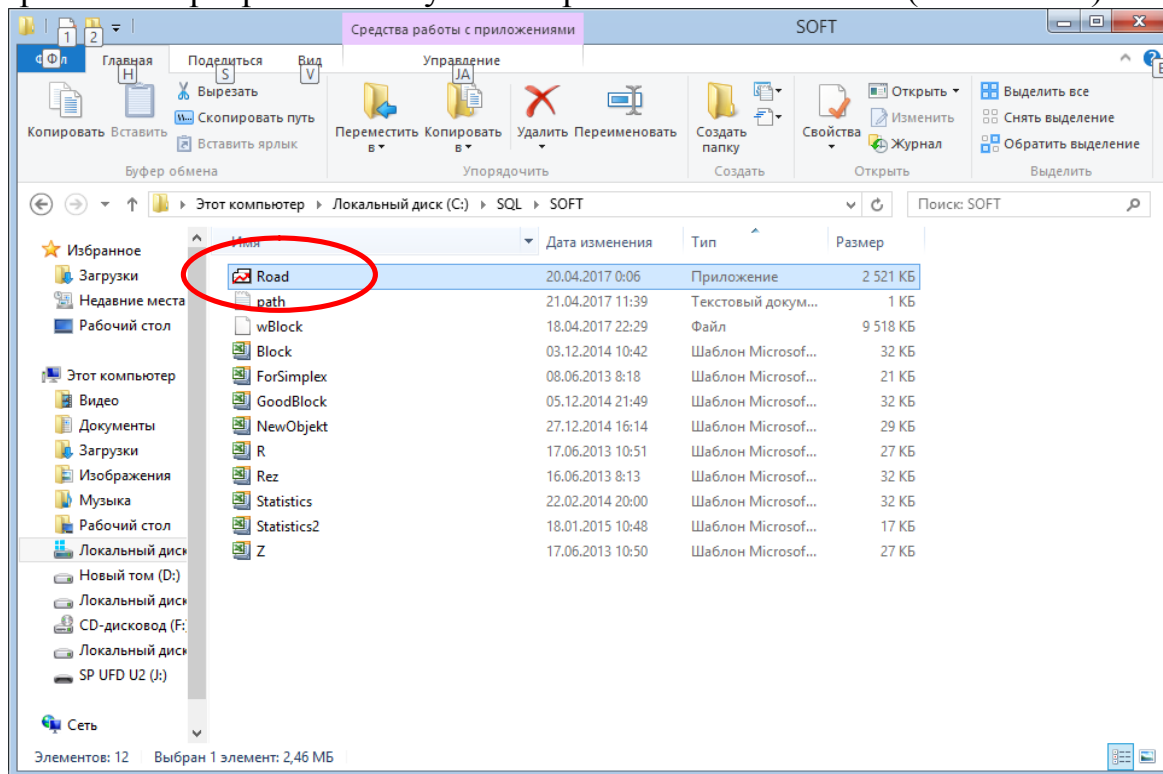


Рис. 20. Запуск приложения Road.exe.

При первом запуске приложения появится предупреждение (см. Рис.21).

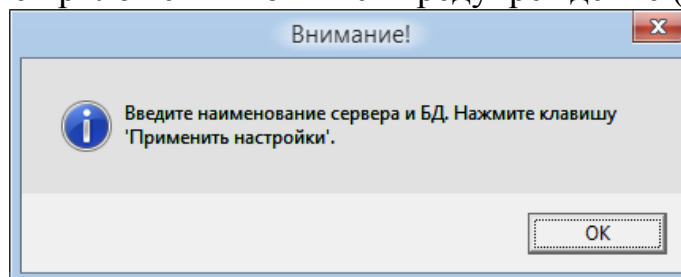


Рис. 21. Предупреждение о необходимости настроек.

Нажимаем кнопку **OK**. Откроется главная форма настройки программы с полями для ввода наименований сервера и БД (см. Рис. 22).

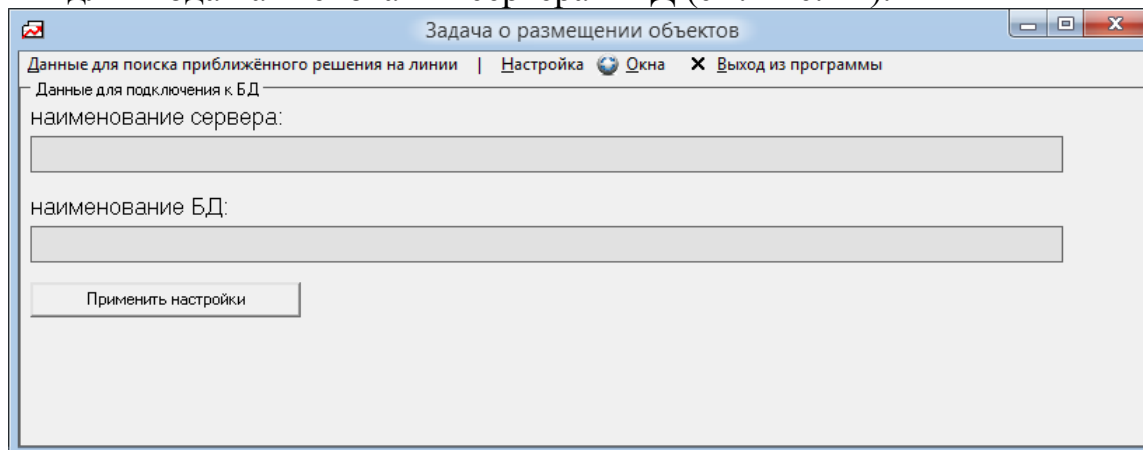


Рис. 22. Настройка данных для подключения к БД.

Вводим наименования сервера и БД в соответствующие поля и нажимаем кнопку *Применить настройки* (см. Рис. 23).

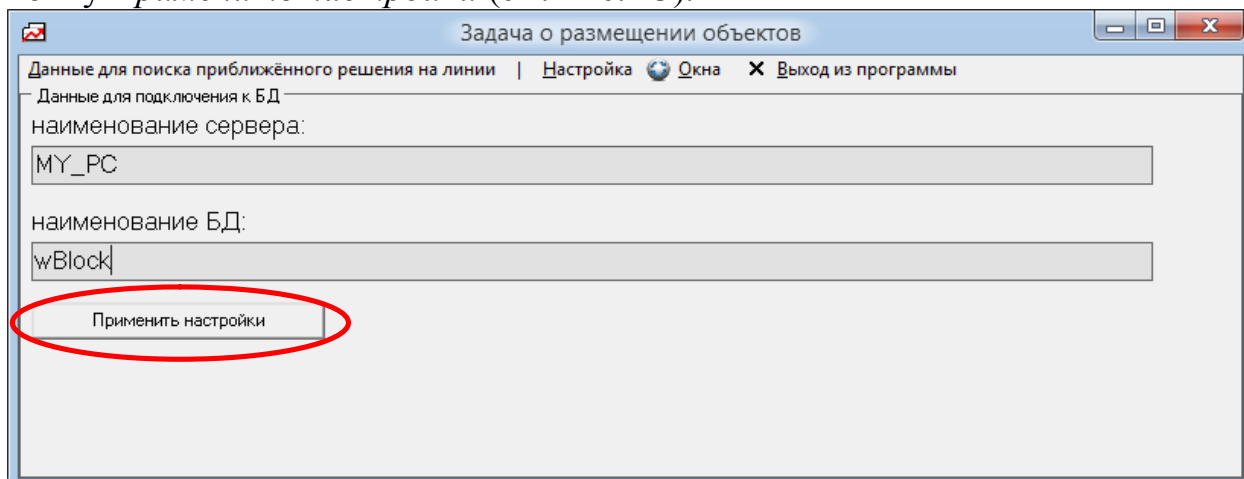


Рис. 23. Ввод наименования сервера и БД.

Появится информационное окно, в котором нажимаем кнопку *OK* (см. Рис. 24).

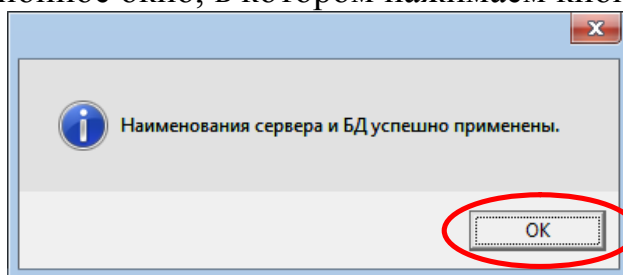


Рис. 24. Информационное сообщение.

Таким образом, установлена связь между программой Road.exe и БД wBlock и далее запускается рабочее окно программы.

Важно! После успешного первоначального восстановления базы и установки связи базы с программой имена сервера и БД записываются в файл path.txt, который находится в папке с программой. При последующих запусках программы считывание значений имён сервера и БД происходит из данного файла.

3.3 Работа пользователя с ПО

Для работы с программой запускаем приложение Road.exe. После чего откроется рабочая форма (см. Рис. 25).

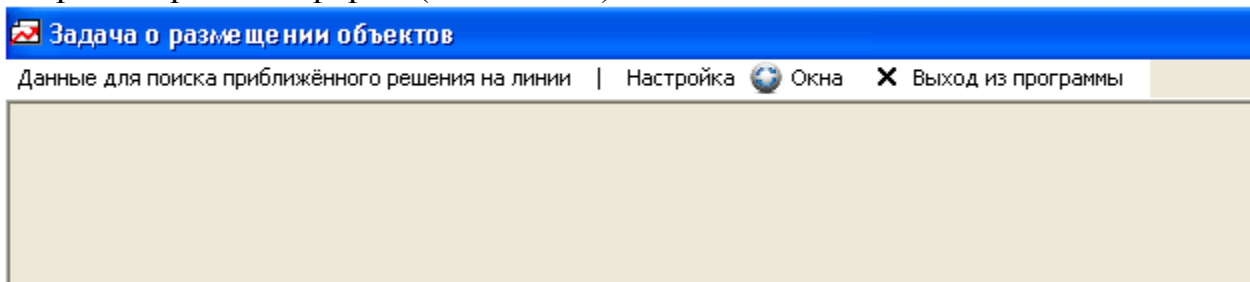


Рис. 25. Рабочая форма программы.

На верхней панели инструментов указаны основные пункты меню. В меню *Настройка* прописываются имена сервера и БД (уже было описано выше),

меню *Окна* – для перехода между рабочими областями. В меню *Данные для поиска приближенного решения на линии* содержатся два пункта: *Справочник задач для поиска приближенного решения на линии* и *Решение алгоритмом поиска приближенного решения на линии* (см. Рис. 26).

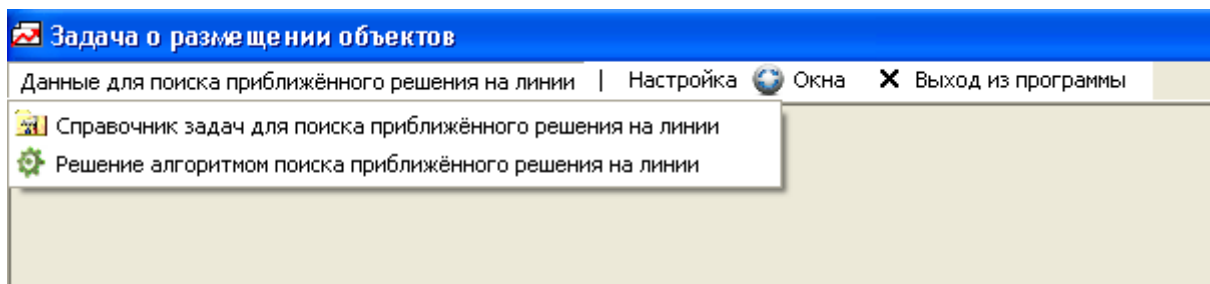


Рис. 26. Меню *Данные для поиска приближенного решения на линии*.

3.3.1 Справочник задач для поиска приближенного решения на линии

При выборе пользователем пункта *Справочник задач для поиска приближенного решения на линии* открывается окно справочника, где указан список всех тестируемых задач (см. Рис. 27). Справочник предназначен для ввода, хранения и редактирования списка всех тестируемых задач.

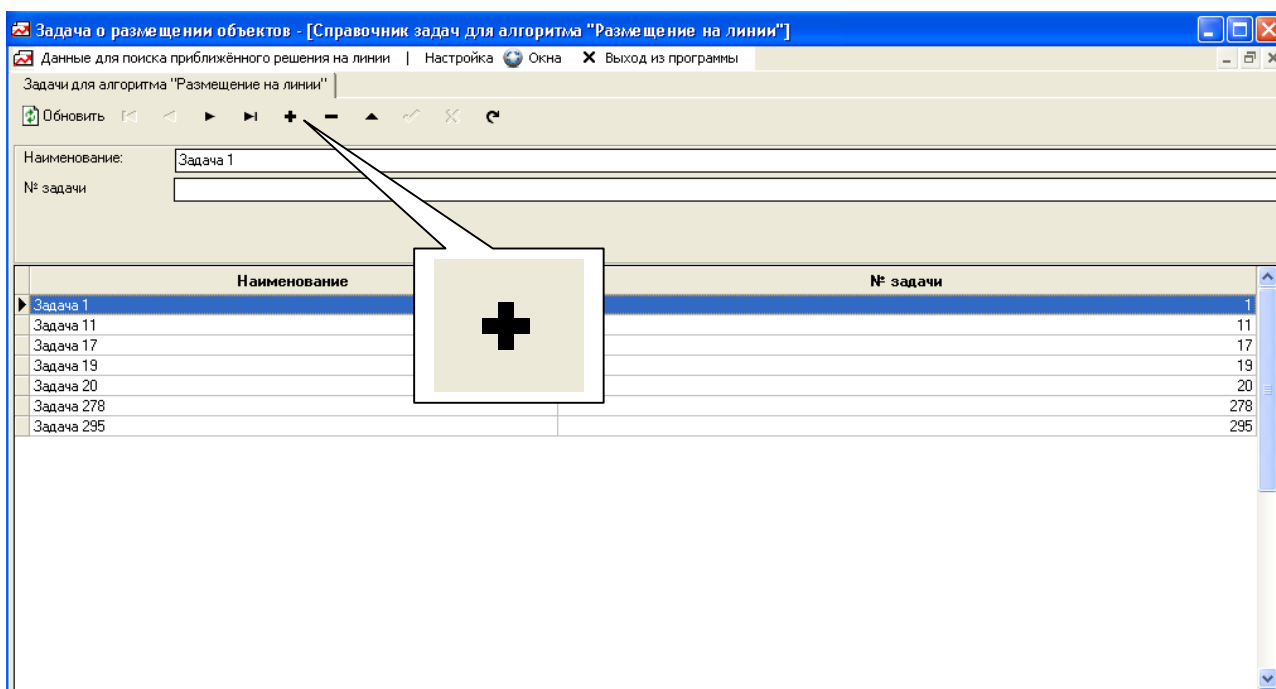


Рис. 27. Меню *Справочник задач для поиска приближенного решения на линии*.

На верхней панели инструментов справочника расположены кнопки ввода, редактирования, удаления записи для соответствующей задачи, обновления списка задач.

При вводе новой задачи нужно нажать кнопку **+**, затем ввести в соответствующие поля наименование задачи и ее номер (Рис. 28):

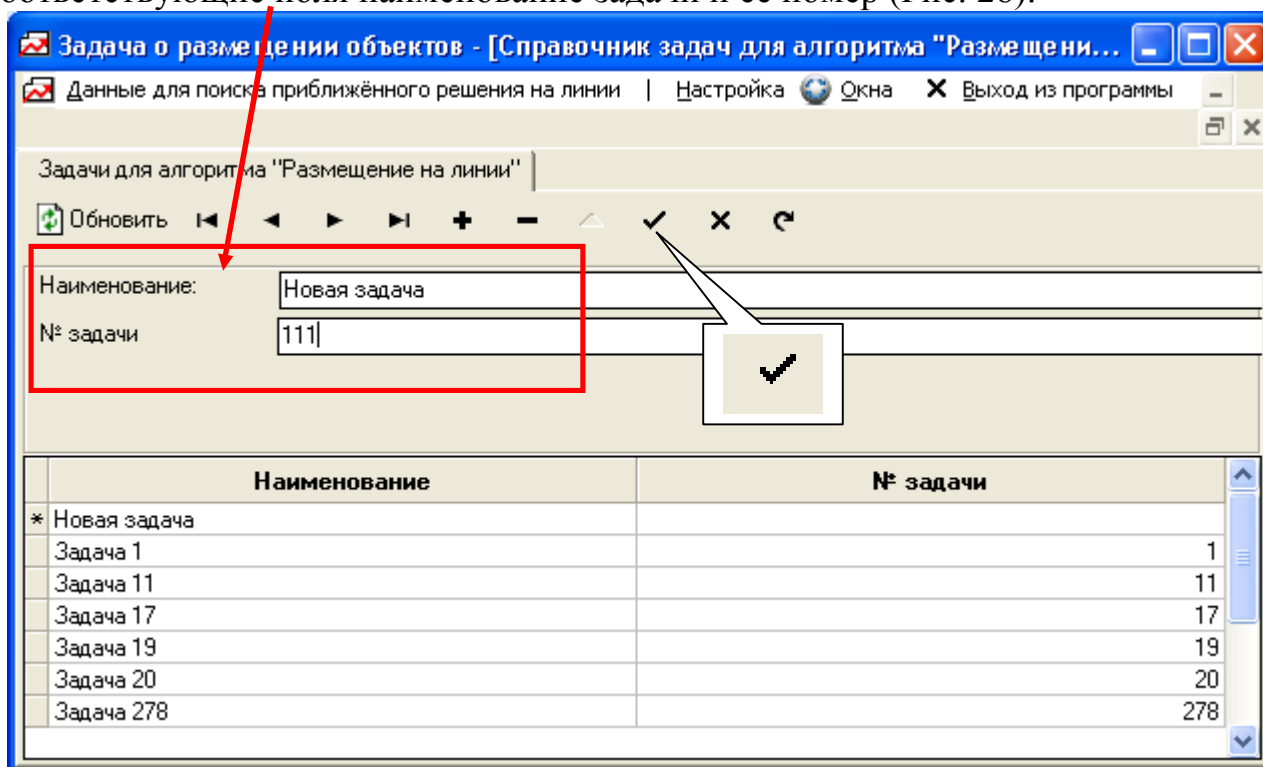


Рис. 28. Поля для ввода новой задачи в справочник.

Далее, нажать кнопку **✓** и вводимая задача появится в списке:

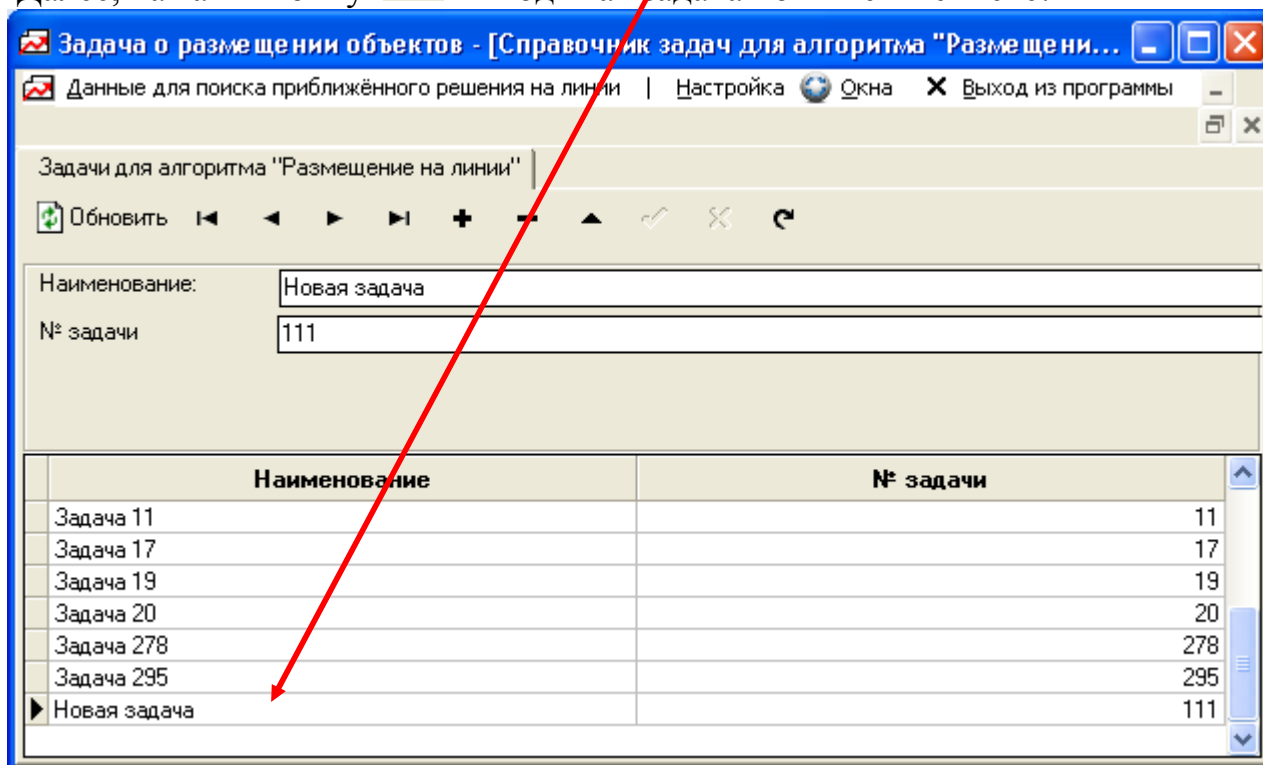


Рис. 29. Ввод новой задачи в Справочник задач для поиска приближенного решения на линии.

3.3.2 Решение алгоритмом поиска приближенного решения

В пункте *Решение алгоритмом поиска приближенного решения* содержится форма с вкладками *Начальные данные* и *Статистика* (см. Рис. 30).

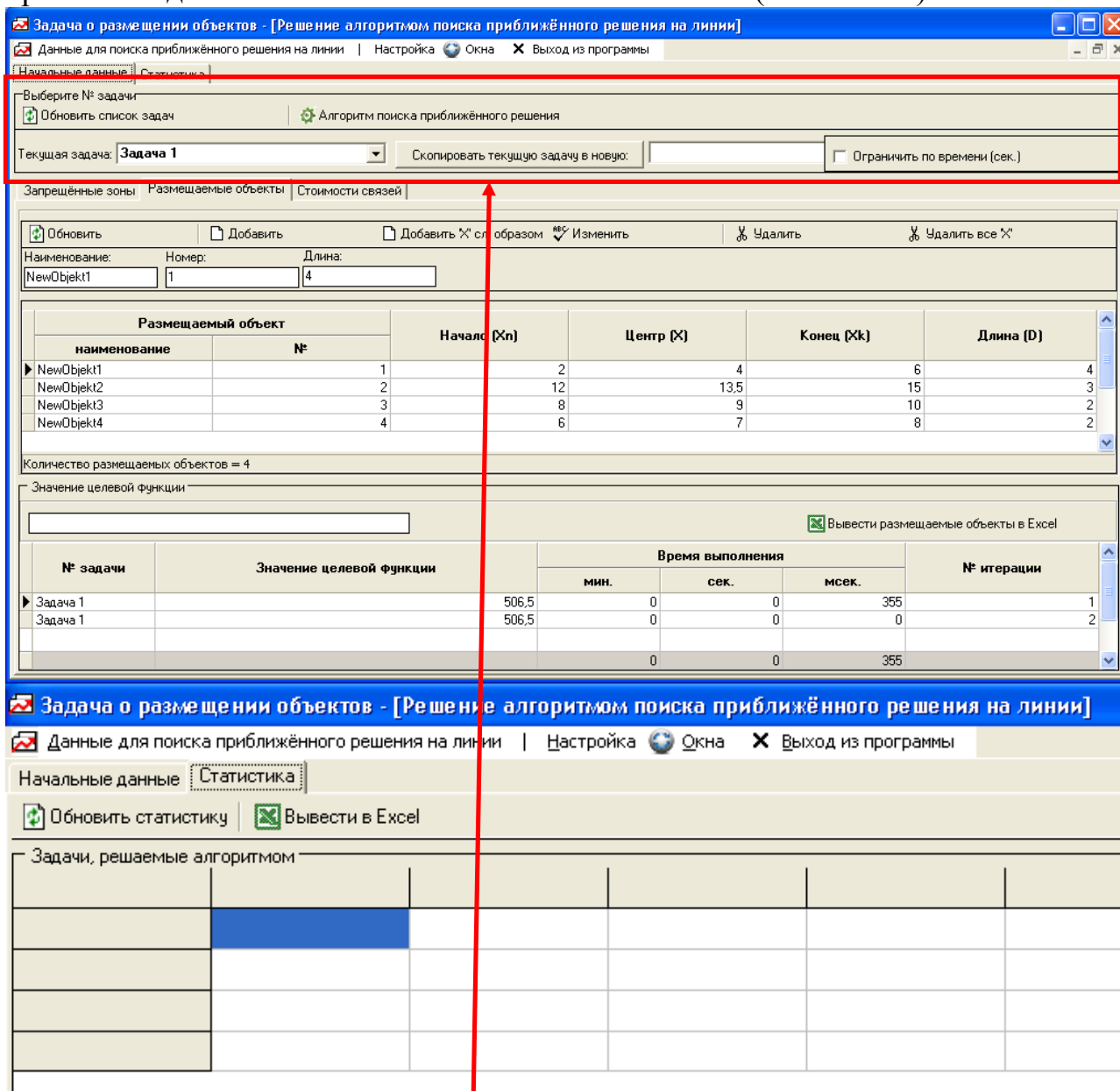


Рис. 30. Меню *Решение алгоритмом поиска приближенного решения*.

Во вкладке *Начальные данные* содержатся все инструменты, с помощью которых вводятся начальные данные и решается задача. К начальным данным относится информация о координатах центров, длинах запрещенных зон, о числе и длинах размещаемых объектов, об удельных стоимостях связей размещаемых объектов с зонами и между собой. Также на данной вкладке имеется группа *Выберите № задачи* (см. Рис. 31).

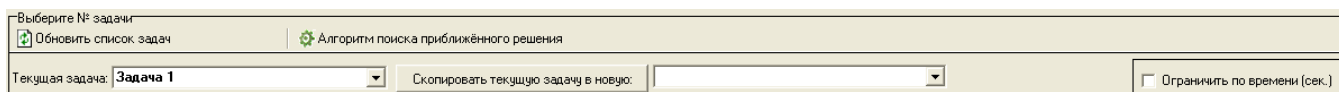


Рис. 31. Группа *Выберите № задачи*.

Данная группа содержит инструменты для перехода от одной задачи *Справочника задач для поиска приближенного решения на линии* к другой, находясь на вкладке *Начальные данные*. Описание справочника см. п. 3.3.1.

Для того, чтобы на вкладке *Начальные данные* отобразить данные по интересующей задаче, нужно выбрать её в выпадающем списке *Текущая задача*. (см. Рис. 32).

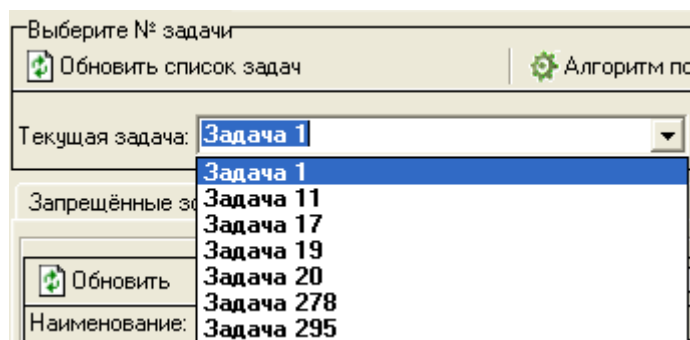
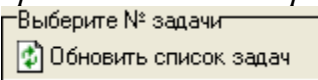


Рис. 32. Выбор задачи из выпадающего списка.

Для копирования данных выбранной задачи, например, *Задачи 1* в *Новую задачу* необходимо создать новую задачу в *Справочнике задач для поиска приближенного решения на линии* (см. Рис. 29). Нажать клавишу

 группы *Выберите № задачи* (см. Рис. 31). Далее, выбираем *Новую задачу* во втором выпадающем списке группы (см. Рис. 33).

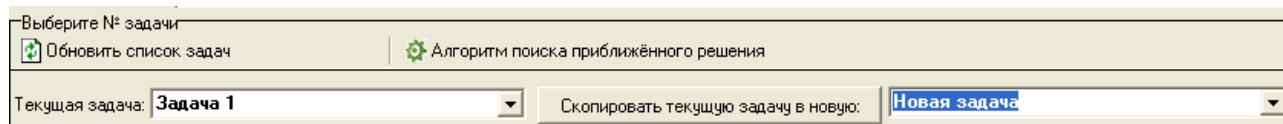
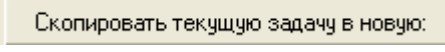
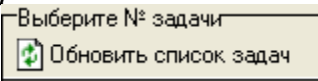


Рис. 33. Выбор *Новой задачи* для копирования в неё текущей.

Нажав клавишу , мы произведём копирование всех данных *Текущей задачи (Задачи 1)*, выбранной в первом выпадающем списке, в *Новую задачу*, выбранной во втором выпадающем списке. Таким образом, мы существенно ускоряем процесс ввода задач для решения.

После ввода новой задачи в *Справочника задач для поиска приближенного решения на линии* нужно на вкладке *Начальные данные* нажать кнопку

 (см. Рис. 30), тогда введенная задача появится в выпадающем списке *Текущая задача*.

После выбора пользователем конкретной задачи из выпадающего списка *Текущая задача*, все данные о выбранной задаче отразятся в одноименных вкладках формы. Информация о запрещенных зонах отразится во вкладке *Запрещенные зоны*, о размещаемых объектах – в *Размещаемые объекты*, стоимости связей – во вкладке *Стоимости связей*. Во всех этих вкладках есть функциональные кнопки ввода, редактирования, удаления и добавления случайным образом соответствующих данных, а также *вывода в Excel* для последующей визуализации.

Важно! Порядок ввода начальных данных: сначала необходимо ввести информацию о запрещённых зонах (наименование, номер, центр, длина) – вкладка *Запрещённые зоны* и размещаемых объектах (наименование, номер, длина) – вкладка *Размещаемые объекты*, а уже потом стоимости связей. Порядок ввода именно таков, так как при генерации стоимостей связей случайным образом нужно знать число запрещённых зон и число размещаемых объектов.

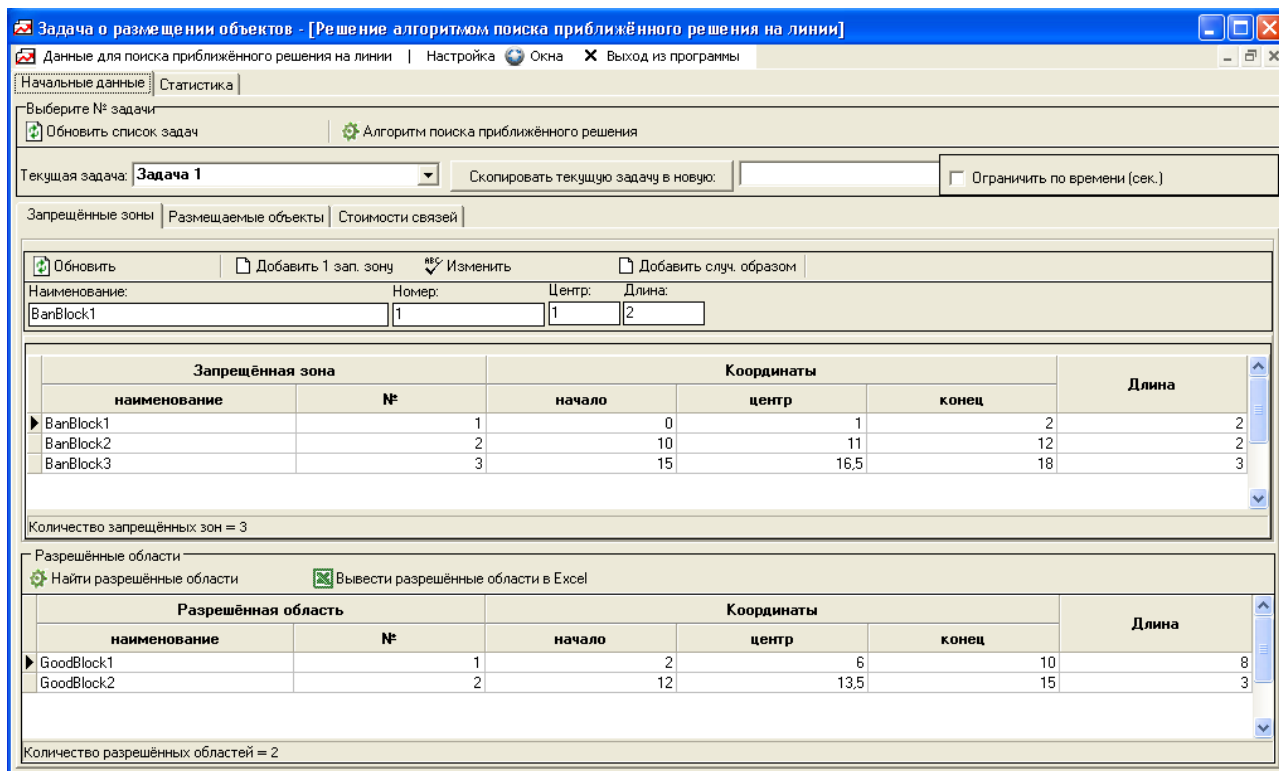


Рис. 34. Вкладка *Запрещённые зоны*.

При вводе запрещенной зоны нужно ввести *наименование, номер*, координаты *центра* и ее *длину* и нажать клавишу . Предусмотрена возможность ввода запрещенных зон случайным образом – клавиша .

Важно! Необходимо помнить, что отрезок, на котором размещаются объекты, по умолчанию должен начинаться с запрещенной зоны и заканчиваться также запрещенной зоной. По введенным запрещенным зонам можно построить разрешенные области (блоки), нажав на одноименные клавиши данной формы, а также вывести их в Excel (см. Рис. 34, Рис. 35).

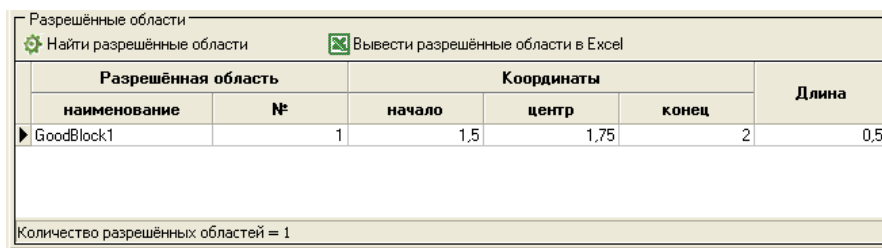


Рис. 35. Группа *Разрешённые области* вкладки *Запрещённые зоны*.

Во вкладке *Размещаемые объекты* вводится информация о размещаемых объектах, а именно *наименование, номер, длина* (см. Рис. 36).

При вводе размещаемого объекта нужно ввести *наименование, номер* и его *длину*. Нажать клавишу . Предусмотрена возможность ввода размещаемого объекта случайным образом – клавиша .

На этой же вкладке будет в последствии выводиться результат решения задачи, в виде значений координат центров размещаемых объектов, а так же значения целевой функции, времени счета, числа итераций – группа *Значение целевой функции* (см. Рис. 36).

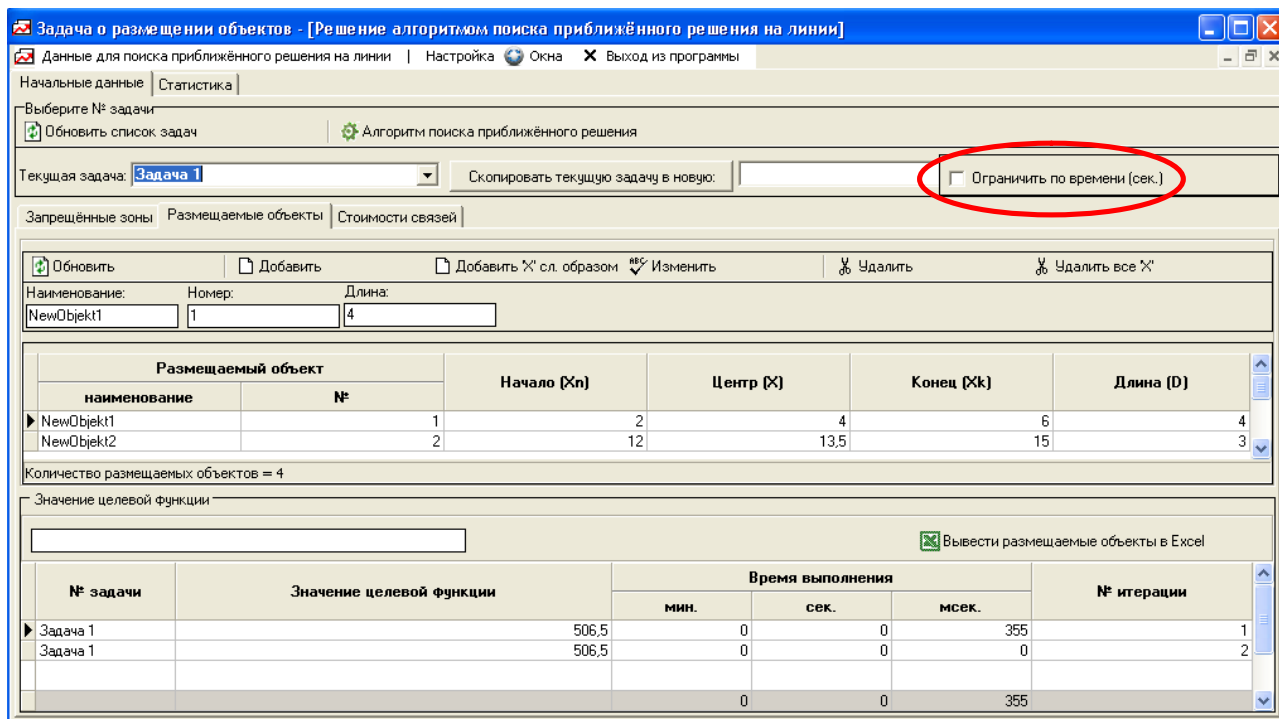


Рис. 36. Вкладка *Размещаемые объекты*.

Во вкладке *Стоимости связей* вводятся данные о стоимости связей размещаемых объектов с запрещёнными зонами (W_{ij}) и между собой (V_{ik}) (см. Рис. 37).

Важно! Значения стоимостей связей должны быть положительными величинами и $v_{ik} = v_{ki}$.

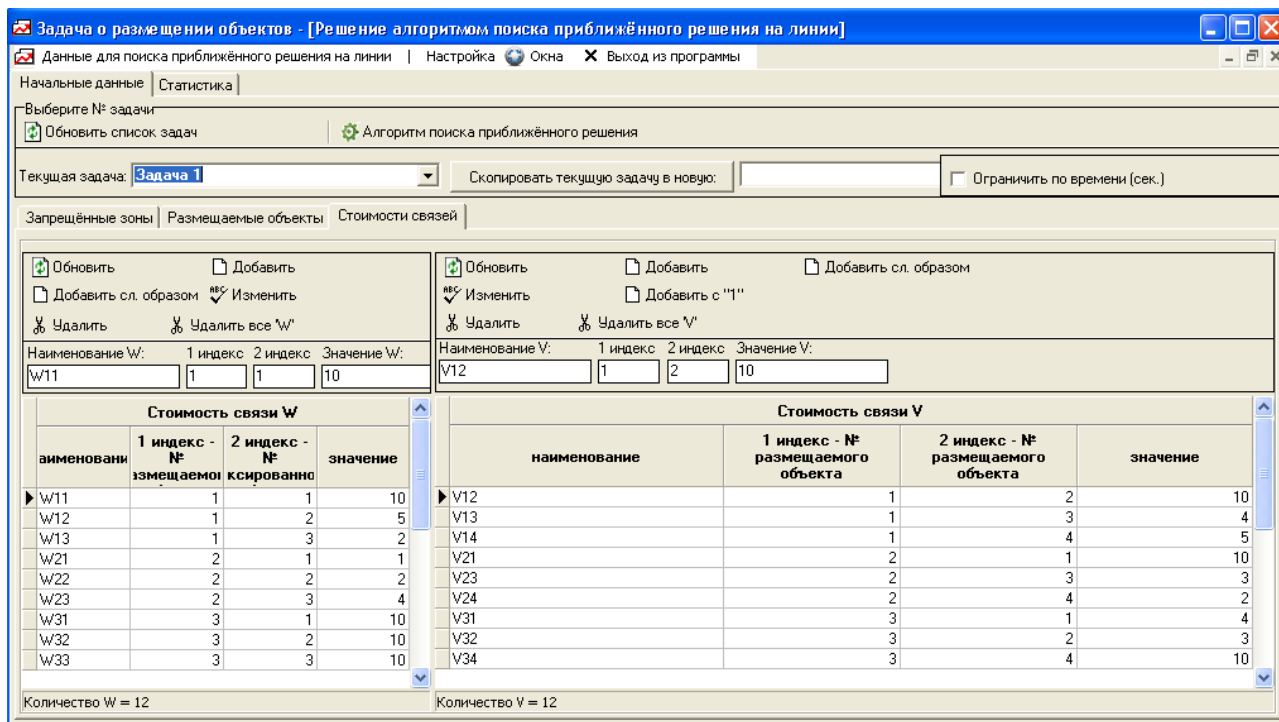


Рис. 37. Вкладка *Стоимости связей*.

После ввода данных о запрещенных зонах, о размещаемых объектах и стоимостях связей для нахождения решения задачи нужно нажать кнопку **Алгоритм поиска приближённого решения**. Как уже говорилось, результат решения выводится во вкладке *Размещаемые объекты*. Предусмотрена возможность вывода в Excel (см. Рис. 36, клавиша **Вывести размещаемые объекты в Excel**). При нажатии на данную клавишу сформируется файл Excel – NewObjekt.xls (см. Рис. 38).

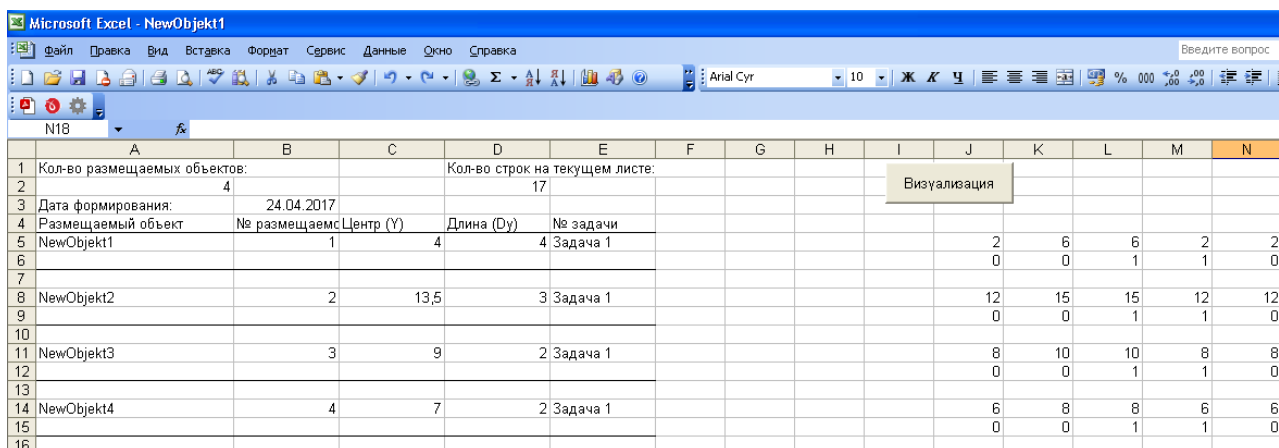


Рис. 38. Вывод решения в Excel.

При выводе в Excel предусмотрена визуализация полученных результатов, для этого реализована клавиша **Визуализация** (см. Рис. 39).

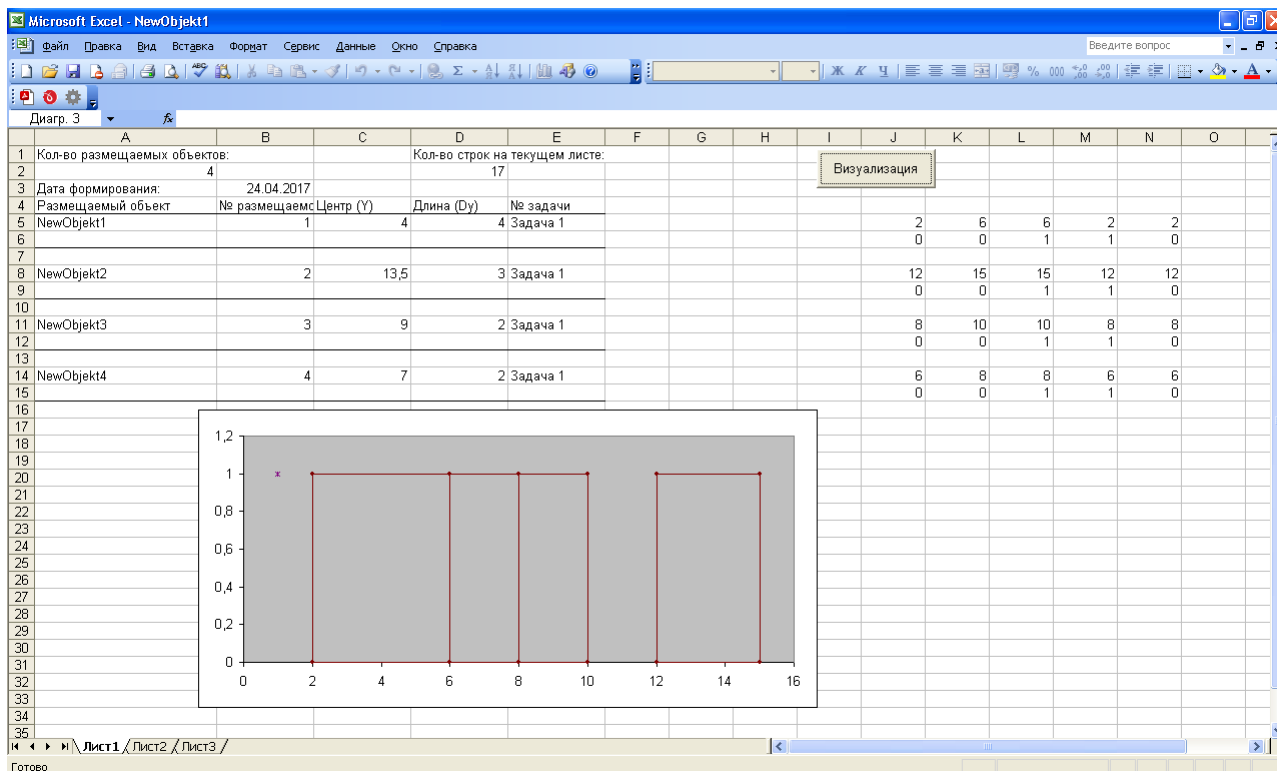
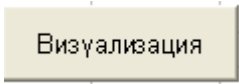
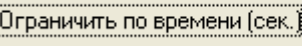
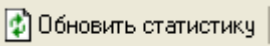
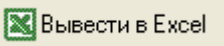


Рис. 39. Визуализация решения в Excel.

При нажатии на клавишу  сформируется диаграмма с найденными размещаемыми объектами (см. Рис. 39).

По умолчанию критерием остановки алгоритма является перебор всех допустимых разбиений объектов по блокам. При необходимости, можно ограничить работу алгоритма по времени, поставив галочку в поле  (см. Рис. 36). В этом случае алгоритм выдаст лучшее найденное решение за указанное время.

Во вкладке *Статистика* рабочей формы пользователь может увидеть результаты тестирования всех задач из справочника. Для этого нужно нажать кнопку  (см. Рис. 40). Также можно все полученные данные вывести в табличную форму в Excel (см. Рис. 41), нажав кнопку 

Наименование задачи	кол-во R	кол-во Z	кол-во X	кол-во W	кол-во V	t выполнения, мин.	t выполнения, сек.	t выпод
Задача 1	2	3	4	12	12	0,0059166666666667	0,355	355
Задача 11	2	3	5	15	20	0,0516833333333333	3,101	3101
Задача 17	3	4	5	20	20	0,1117666666666667	6,706	6706
Задача 19	1	2	12	24	132	0,0476	2,856	2856
Задача 20	4	5	6	30	30	0,0750666666666667	4,504	4504
Задача 278	9	10	10	100	90	0,6267166666666667	37,603	37603
Задача 295	19	20	30	600	870	0,48615	29,169	29169

Рис. 40. Вкладка *Статистика*.

№ задачи	Количество					Время выполнения			Кол-во итераций	Значение целевой функции
	R	Z	X	W	V	мин.	сек.	мсек.		
Задача 1	2	3	4	12	12	0,00591667	0,355	355	2	506,5
Задача 11	2	3	5	15	20	0,05168333	3,101	3101	5	1287
Задача 17	3	4	5	20	20	0,11176667	6,706	6706	11	4070
Задача 19	1	2	12	24	132	0,0476	2,856	2856	2	10736
Задача 20	4	5	6	30	30	0,07506667	4,504	4504	6	7841,5
Задача 278	9	10	10	100	90	0,62671667	37,603	37603	21	13413
Задача 295	19	20	30	600	870	0,48615	29,169	29169	3	259846

Рис. 41. Вкладка *Статистика*. Вывод в Excel.

В режиме онлайн пользователь может увидеть все данные по выбранной задаче из *Справочника задач для поиска приближенного решения на линии*, что удобно для анализа полученных решений и тестирования задач. Возможность визуализации также создает удобства для восприятия результатов решения, так как пользователь может наглядно увидеть запрещенные зоны, разрешенные области и размещаемые объекты для выбранной задачи.

Разработанный алгоритм предназначен для получения проектных решений, например, по размещению единиц технологического оборудования в цехах промышленного предприятия.