

Описание принципов работы генетического оптимизатора и общее описание генетических алгоритмов.

1. Теория оптимизации

Основными понятиями теории оптимизации являются:

- **целевая функция** - оптимизируемая функция, которая численно выражает степень достижения целей оптимизации;
- **решение** - вектор независимых переменных, компоненты которого – переменные оптимизации;
- **пространство оптимизации** - пространство задач оптимизации, размерность которого равна размерности вектора переменных оптимизации. В некоторых случаях возможно оценить количество состояний задачи и, следовательно, оценить трудоемкость задачи оптимизации.

Упрощенно любую технологию математической оптимизации можно представить следующим образом. Выбирается целевая функция. Строится вектор переменных оптимизации, который влияет на целевую функцию. Затем варьированием переменных по какому-либо закону находится или экстремум функции или иной критерий (попадание в диапазон, не превышение заданной величины и т.д.).

В зависимости от задачи определяется способ оптимизации. Возможными классическими способами решения оптимизационных задач для данной типа являются методы перебора и методы линейного программирования. Сложность проблемы может потребовать постановки обобщенных, нелинейных задач, использования целевых функций общего вида и, как следствие, использование методов нелинейного программирования.

Задача **математического программирования (линейного или нелинейного)**, ставится, если математические функции, входящие в задачу оптимизации записываются в явном аналитическом виде. В противном случае, если функции не удастся записать аналитически (они заданы в виде компьютерных программ, имитационных моделей, человеко-машинных процедур или как выход реальной экономико-производственной системы), то ставится задача **оптимизации с неявными функциями – поисковая задача оптимизации**.

Поисковые задачи возникают и в том случае, если на систему наложены многочисленные обратные связи, нелинейные и стохастические.

В последние годы, с появлением мощных вычислительных средств были разработаны новые эффективные способы оптимизации. Это большое количество вариантов нейросетевых алгоритмов, исключительно эффективные методы “муравьиных колоний”, эволюционные и генетические алгоритмы и т.п.

1.1 Методы полного перебора

Методы полного перебора имеют существенные ограничения по применимости и могут быть использованы только в системах с небольшим количеством состояний. Допустим, целевая функция зависит от 10 дискретно изменяющихся параметров. Если каждый параметр может принимать только 2 значения 0 и 1, количество состояний системы равно $2^{10}=1024$. Реальные системы с сотнями параметров (в том числе и непрерывно изменяющимися) могут потребовать перебора миллиардов вариантов, что превышает возможности любых компьютеров.

1.2 Математическое программирование

В простейшем случае линейных ограничений и независимых проектов может применяться метод линейного программирования. Тогда задача оптимизации инвестиционных проектов выглядит следующим образом:

Ограничения : $\sum_{i,s} X_{i,s} * C_{i,s,t} \leq C_t$	-сумма расходов на принимаемые проекты в году t не должна превышать установленного лимита
$\sum_s X_{i,s} \leq 1$	-существует единственная дата начала реализации проекта
Целевая функция: $\sum_s X_{i,s} * NPV_{i,s} \Rightarrow \max$	-максимизация NPV портфеля проектов
Где $C_{i,s,t}$ – затраты на i-й проект в году t ; s – год начала реализации проекта $X_{i,s} = 0$ или 1 – принятие//отклонение i-го проекта в году s	

В представленной выше схеме учитываются только ежегодные лимиты инвестиционных расходов, однако в общем виде задача может включать множество дополнительных ограничений и корпоративных ориентиров.

При учете единственного ограничивающего фактора – лимита инвестиционных расходов в плановом году – задача оптимизации упрощается до ранжирования проектов по PI и отбора наилучших проектов до тех пор, пока итоговый набор удовлетворяет ограничению по объему затрат.

Как было отмечено выше, методы математического программирования не могут быть применены, если функции входящие в задачу нелинейные, неявно заданы или ставится задача сложной динамической оптимизации.

1.3 Модифицированные генетические алгоритмы

В случае очень сложных задач, в которых присутствуют сложные нелинейные зависимости и обратные связи, использование методов математического программирования не дает гарантированного оптимального решения, либо вообще неприменимо. В случае, если целевая функция имеет несколько экстремумов эти методы не гарантируют нахождения глобального максимума, а найденный локальный максимум может привести к принятию недостаточно эффективного решения. Поэтому в методике оптимизации рекомендуется

применения **«генетических» алгоритмов**. Терминология этих методов взята из постановок задач эволюционной биологии.

Вектор независимых переменных понимается как так называемая «хромосома», определяющая поведение «индивидуума» значение целевой функции. В нашем случае набором «хромосом» будет набор потенциально правильных (с точки зрения увеличения капитализации при соблюдении ограничений) проектов.

С «хромосомами» могут проводиться следующие действия:

- **«скрещивание - кроссинговер»** - обмен участков «хромосом»;
- **«мутация»** - малое случайное изменение одного или нескольких переменных.

Получившиеся в результате «индивидуумы» оцениваются с точки зрения эффективности пакета инвестиционных предложений, который они описывают. Эффективные «индивидуумы» участвуют в дальнейшем рассмотрении, неэффективные отбрасываются.

Таким образом, «мутации» позволяют выявить новые (скрытые) свойства всего набора решений, скрещивание закрепляет полезные свойства части набора. Отбрасывание («естественный» отбор) позволяет достичь соответствия критерию оптимизации – достижения максимальной эффективности пакета проектов.

При этом для решения задачи выбора между альтернативными (эффективными, но слабо отличающимися по рентабельности) проектами в условиях системы конкурентных ограничений необходимо использовать модифицированный генетический алгоритм, который позволяет учитывать средневзвешенное расстояние характеристик оптимального решения от границы области допустимых значений (по всем типам ограничений). Возможно также введение дополнительных целевых функций – вектора критериев оптимизации, возможно имеющих различные свойства (экономические, технологические, психологические и т.д.)

При этом весовые коэффициенты отражают степень предпочтения решения. В мировой литературе показано, что для подобных сложных, многокритериальных задач оптимизации «генетические алгоритмы» демонстрируют наибольшее отношение эффективности решения к трудоемкости его получения. по сравнению с традиционными методами.

Реализация генетических алгоритмов возможна в среде Powersim. Встроенный генетический оптимизатор является чрезвычайно мощным и гибким средством решения сложных оптимизационных задач.

1.4 Powersim Studio 2005.

Программное обеспечение Powersim Studio 2005 - это инструмент визуального моделирования позволяющий реализовать *математическую модель* с помощью потокового графа, отражающего изменение характеристик модели (реальные цифры) *в сжатом времени*. Такой подход позволяет описать систему высокой сложности на визуальном языке, что автоматически делает ее *самодокументируемой* (в отличие от Excel).

Программная интегрированная среда PowerSim 2005 позволяет моделировать реальные процессы, протекающие в бизнес - системах, в динамике, используя наглядный и простой интерфейс. Удобный интерфейс программы позволяет работать с конструктором модели на абстрактном уровне: осуществлять добавление/удаление новых объектов и переменных, изменять

связи между ними. И вся эта процедура не требует от пользователя знания внутреннего языка системы.

Используемая система моделирования PowerSim позволяет первоначально представить реальную модель в графическом виде, задав центры прибыли и затрат и описав связи между ними. Далее, путем использования визуальных средств системы PowerSim, можно задать временной шаг прогона модели и определить промежуток прогнозирования. Разработчик может учитывать причинно-следственные связи между объектами модели посредством возможностей интегрированной среды. Построение модели может быть осуществлено иерархическим образом сверху-вниз. При этом, имеется возможность отдельно проанализировать работу составляющих систему модулей. Программа позволяет моделировать реальную систему, используя несколько логических объектов, отражающих основные связи, возникающие в реальности: диаграммы взаимосвязей между переменными среды, объекты обратной связи, линии задержки. Логической основой этих объектов является взаимосвязь двух понятий: уровней и потоков, отражающих динамическую природу экономических переменных.

Последовательность операций динамического моделирования представлена на рис.1-1. На основе представлений о работе реального предприятия строится когнитивная модель, отражающая причинно-следственные связи. Затем строится математическая модель, которая реализуется средствами визуального программирования в среде PowerSim Studio 2005. К построенной таким образом визуальной модели можно применять различные методы (оптимизация, анализ рисков и т.д.), задавать вопросы, проигрывать различные сценарии поведения и отклика на внешние факторы. Система имеет достаточно развитые средства для визуализации результатов – построение таблиц, графиков, элементов управления.

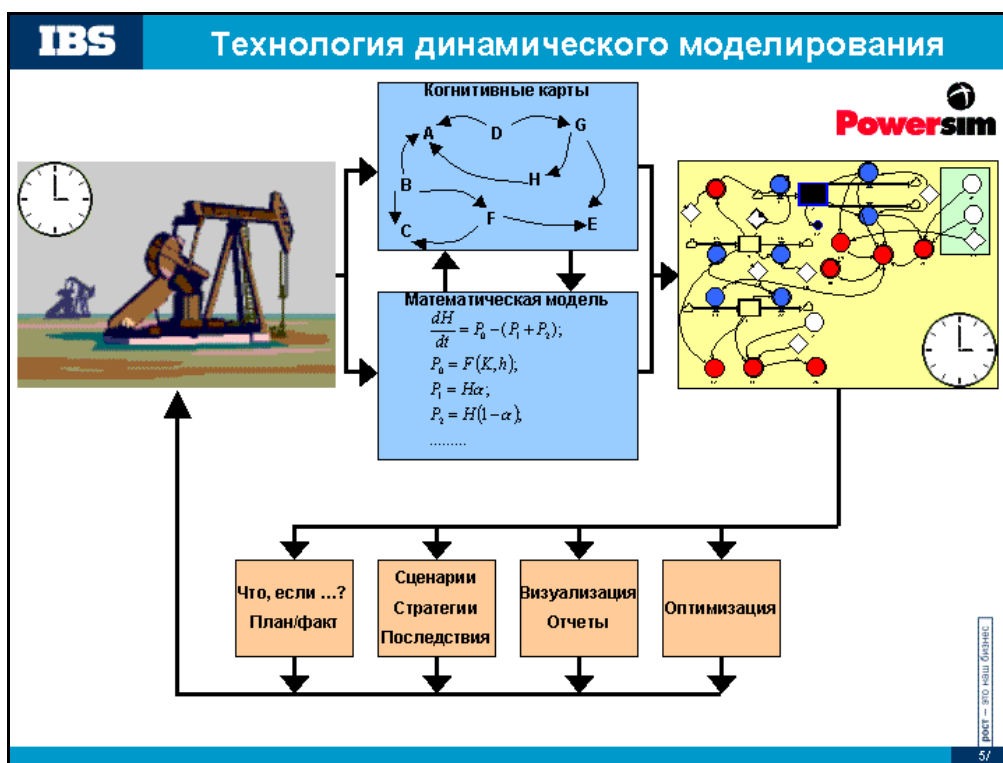


Рис. 1-1. Технология динамического моделирования.

Динамическая программная модель позволяет провести анализ возможных сценариев развития модели производства, учитывая их в одной модели при достижении определенными переменными краевых значений.

Таким образом, можно выделить основные преимущества моделирования в среде PowerSim 2005:

- визуализация бизнес-процессов на функционально-стоимостном уровне;
 - учет временных лагов;
 - учет вероятностей и рисков;
 - автоматизированный учет ограничений;
 - наличие перекрестных ссылок;
 - удобный для пользователя интерфейс.
-

1.5 Технология оптимизации в PowerSim (PowerSim Solver)

Для решения оптимизационных задач в рассматриваемой модели предлагаются современные методы генетической оптимизации (генетические алгоритмы). Математически они относятся к классу задач эволюционного программирования, активно разрабатываются в последние годы и доказали свою эффективность для большого количества практических задач. Краткое описание генетической оптимизации дано ниже.

Встроенный оптимизатор PowerSim (PowerSim Solver 2.5) построен именно на таких методах. Он прост и гибок в управлении и вместе с тем, обладает достаточной эффективностью для решения различных задач.

Важным достоинством решений на базе оптимизатора PowerSim является возможность формирования взвешенного вектора оптимизационных критериев. Причем природа критериев может быть разной. Можно потребовать максимальной доходности при минимальной длительности, например. При этом относительная значимость критериев также может быть выбрана.

Вместе с тем, возможности оптимизатора ограничены, во-первых, это интерактивная программа, требующая ручного ввода параметров, во-вторых, его быстродействие может оказаться недостаточным для некоторых постановок задач. Существуют библиотеки программ для генетической оптимизации, которые можно использовать в PowerSim через механизмы SDK. У разработчиков есть и свои оригинальные алгоритмы, которые могут быть использованы вместе с PowerSim в случае необходимости и развития задачи.

2. Генетические алгоритмы.

2.1 Естественный отбор в природе

Теория эволюции утверждает, что каждый развитие биологических видов идет в сторону наилучшего приспособления к окружающей среде. Можно сказать, что эволюция - это процесс оптимизации живых организмов.

Основной механизм эволюции - **естественный отбор**. При отборе более приспособленные особи имеют больше возможностей для выживания и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом благодаря передаче генетической информации, потомки наследуют от родителей их основные качества. Следовательно, потомки сильных индивидуумов также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены многих поколений средняя приспособленность особей данного вида возрастает.

Наследственная информация закодирована в **молекуле ДНК**, находящейся в **хромосоме** ядра клетки. Каждое качество особи кодируется определенной частью хромосомы, которая называется **геном** этого свойства. Все свойства особи (**фенотип**) определяются ее **генотипом**.

При размножении животных происходит слияние двух родительских половых клеток. Их ДНК взаимодействуют, образуя ДНК потомка. Основным способом взаимодействия - **кроссовер (скрещивание)**. При кроссовере ДНК родителей делятся на две части, а затем обмениваются своими половинками.

При наследовании возможны **мутации** из-за радиоактивности или других влияний, в результате которых могут измениться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Если эти новые свойства полезны, они, скорее всего, сохраняются в данном виде - при этом произойдет скачкообразное повышение приспособленности вида.

2.2 Методы оптимизации.

Пусть дана некоторая сложная функция (**целевая функция**), зависящая от нескольких переменных, и требуется найти такие значения переменных, при которых значение функции максимально.

Целевая функция может иметь очень большое количество аргументов и еще большее количество значений. Ее рельеф в таком многомерном пространстве может быть очень сложным. Даже в простых задачах оптимизации часто невозможно получить приемлемое решение методами, основывающимися на **переборе** возможных решений. Методы, использующие анализ рельефа функции (различные варианты **градиентных методов**) могут работать значительно быстрее, но не гарантируют нахождение требуемых экстремумов. Процесс оптимизации может остановиться вблизи локального максимума.

Требуется объединение метода анализа всей функции, которое дает перебор и эффективное нахождение локальных максимумов, которое дают градиентные методы. Генетический алгоритм представляет собой именно такой комбинированный метод. Механизмы скрещивания и мутации в каком-то смысле реализуют переборную часть метода, а отбор лучших решений - градиентный спуск. На рисунке показано, что такая комбинация позволяет обеспечить устойчиво хорошую эффективность генетического поиска для любых типов задач.

Генетический алгоритм (рис. 2-1)- это простая модель эволюции в природе, реализованная в виде компьютерной программы. В нем используются как аналог механизма генетического наследования, так и аналог естественного отбора. При этом сохраняется биологическая терминология в упрощенном виде.

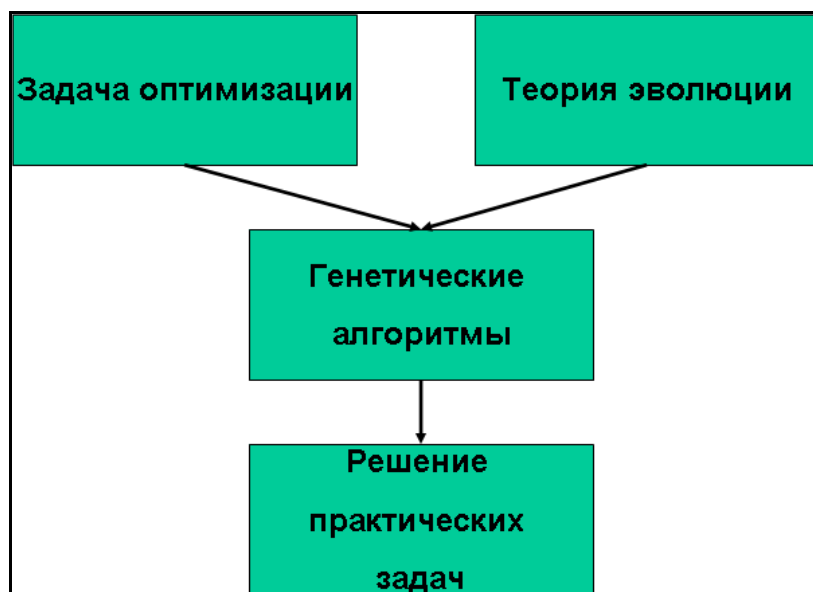


Рис. 2-1. Генетический алгоритм

2.3 Пример реализации генетической оптимизации

Попытаемся решить транспортную задачу, применяя эволюционные способы оптимизации. Будем рассматривать каждый вариант маршрутизации (набор значений загрузки транспортных линий) как индивидуума, а некоторый параметр, например транспортные издержки - как приспособленность этого индивидуума. Тогда в процессе эволюции (если мы сумеем его организовать) приспособленность индивидуумов будет возрастать, а значит, будут появляться все более и более дешевые варианты доставки. Остановив эволюцию в некоторый момент и выбрав самого лучшего индивидуума, мы получим достаточно хорошее решение задачи.

Чтобы смоделировать эволюционный процесс, сгенерируем вначале случайную популяцию - несколько индивидуумов со случайным набором хромосом (числовых векторов загрузки транспортных линий). Генетический алгоритм имитирует эволюцию этой популяции как циклический процесс скрещивания индивидуумов и смены поколений.

Жизненный цикл популяции - это несколько случайных скрещиваний (посредством кроссовера) и мутаций, в результате которых к популяции добавляется какое-то количество новых индивидуумов. Отбор в генетическом алгоритме - это процесс формирования новой популяции из старой, после чего старая популяция погибает. После отбора к новой популяции опять применяются операции кроссовера и мутации, затем опять происходит отбор, и так далее (рис. 2-2).

Таким образом, модель отбора определяет, каким образом следует строить популяцию следующего поколения. Как правило, вероятность участия индивидуума в скрещивании берется пропорциональной его приспособленности. Часто используется так называемая **стратегия элитизма**, при которой несколько лучших индивидуумов переходят в следующее поколение без изменений, не участвуя в кроссовере и отборе. В любом случае каждое следующее поколение будет в среднем лучше предыдущего. Когда приспособленность индивидуумов перестает

заметно увеличиваться, процесс останавливают и в качестве решения задачи оптимизации берут наилучшего из найденных индивидуумов.

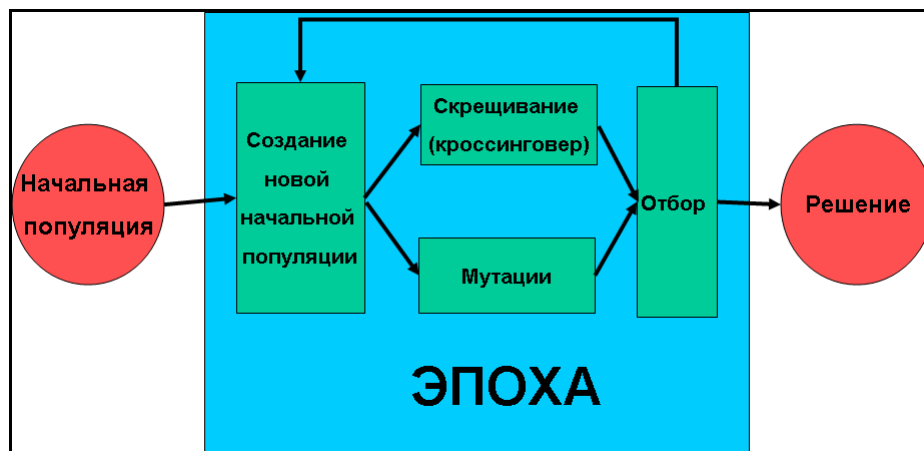


Рис. 2-2. Цикл работы генетического алгоритма.

В итоге, мы находим индивидуума – вектор загрузки транспортных линий при котором транспортные издержки минимальны. В качестве критерия оптимизации могут выступать и другие функции, и их комбинации.

2.4 Выводы:

- Наследственная информация содержится в геноме (хромосомах)
- Фенотип (целевая функция) определяется генотипом (набором параметров).
- Мутации порождают новые варианты.
- Кроссинговер комбинирует существующие варианты.
- Мутации и кроссинговер решают задачу комбинаторного перебора.
- Отбор оставляет жизнеспособные особи.
- Отбор решает задачу градиентного спуска.
- Эволюция решает задачу многопараметрической оптимизации.

Итак, если на некотором множестве задана сложная функция от нескольких переменных, то генетический алгоритм - это программа, которая за разумное время находит точку, где значение функции достаточно близко к максимально возможному. Выбирая приемлемое время расчета, мы получим одно из лучших решений, которые вообще возможно получить за это время.

3. Оптимизационный модуль PowerSim.

3.1 Введение.

Встроенный в PowerSim оптимизатор, использующий генетические алгоритмы, является мощным и удобным в работе модулем, позволяющим находить решение уже сформулированной и построенной задачи, удовлетворяющим определенным критериям. Большое количество настраиваемых пользователем параметров делает его очень гибким и применимым к большому классу предметных задач.

Как уже было отмечено выше, генетический оптимизатор не гарантирует нахождения глобального экстремума целевой функции. Во многом, нахождения оптимума зависит от понимания исследователем предметной задачи. Необходимо грамотное построение комбинаций ограничений и варьируемых параметров – решений. Возможно, потребуется построений целого набора требований к процессу оптимизации и параметров, определяющих сам этот процесс.

Реализация генетических алгоритмов в PowerSim как рядом общих черт для таких программ, так и содержит ряд специфических особенностей (Рис. 3-1).

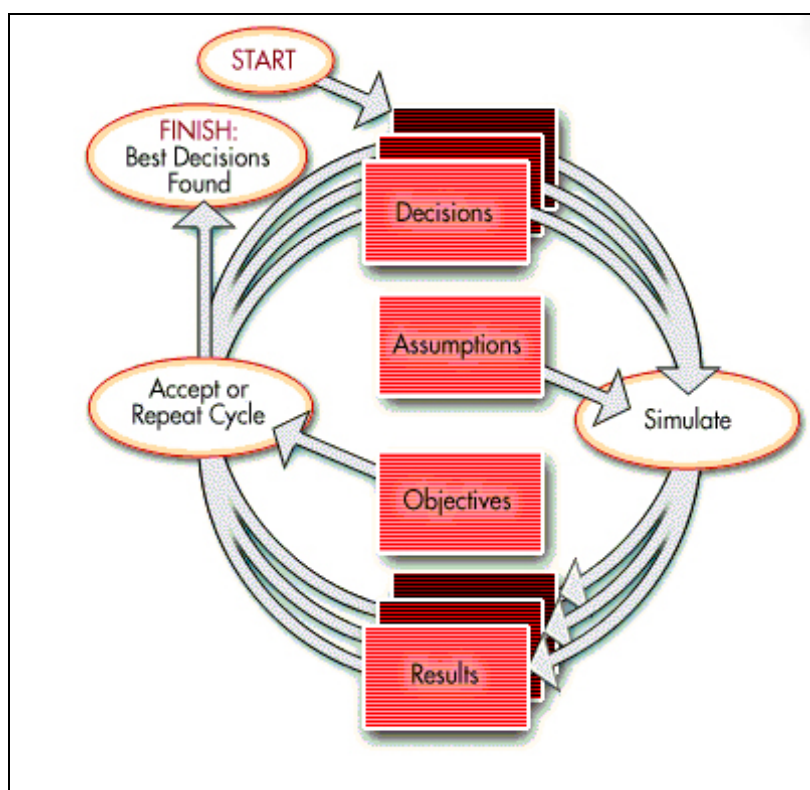


Рис. 3-1. Генетические алгоритмы в PowerSim.

Целевая функция задачи зависит от вектора переменных (**Decision**).

Варьируя величины компонент этого вектора и используя некоторые предположения (**Assumptions**) можно получать в моделировании (**Simulate**) значения целевой функции (**Results**).

Далее, все решения проверяются на соответствие оптимизационному критерию (**Objectives**).

Наиболее удачные решения используются как родительские для следующего поколения (**Accept for Repeat Cycle**).

Весь процесс оптимизации начинается с некоторых начальных предположений (**START**) и заканчивается через неопределенное количество эпох – поколений при выполнении некоторого критерия оптимизации (**FINISH: Best Decisions Found**).

К отличительным особенностям оптимизатора PowerSim относится его способность работы с многомерными переменными, что, впрочем, вытекает из возможностей всего пакета. Работа пакета происходит в интерактивном режиме, но большая гибкость задания параметров как в оптимизаторе, так и в структуре основной задачи позволяет решать достаточно сложные предметные задачи. Весьма ценным свойством является возможность задания взвешенного вектора целевых функций. Тем не менее, невозможность запускать оптимизатор в командном режиме или использовать его как вызываемую из основной программы функцию, снижают его функциональность и гибкость. Но в этом случае, как и случае необходимости реализации более сложных оптимизационных алгоритмов (нейросетевых, “методе муравьиных колоний” и т.д.) возможно программирование на языках высокого уровня и встраивание этих программ в пакет PowerSim средствами Studio SDK 2005.

3.2 PowerSim Solver 2.5 (Рис. 3-2). Версия и разработчики.

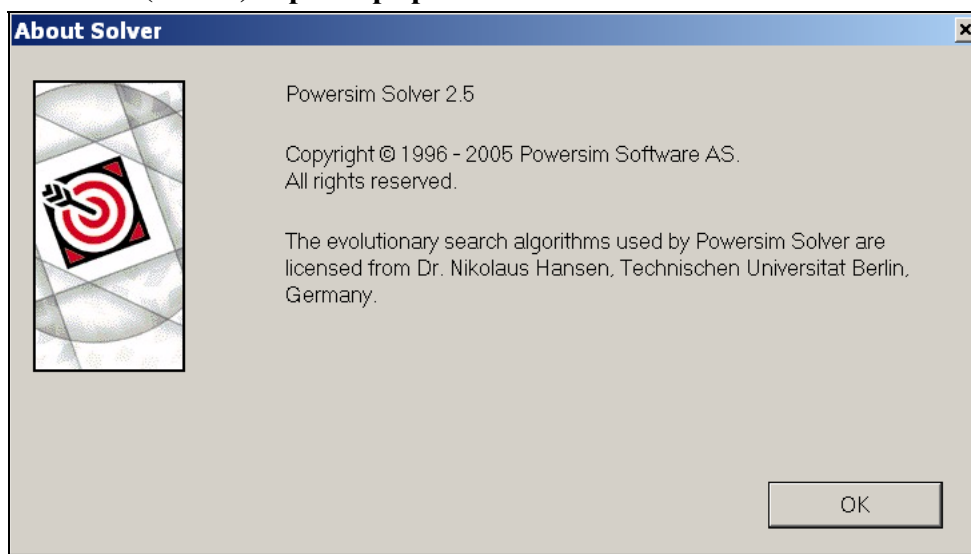


Рис. 3-2. Оптимизатор PowerSim

Более детально работа с оптимизатором в пакете **PowerSim** рассмотрена ниже.

3.3 Окно Analysis Variables

Работа с оптимизатором начинается с подблока **Analysis Variables**. Выбрав этот пункт, далее, в открывшемся правом окне будем задавать оптимизационные переменные и их параметры (Рис. 3-3).

В этом окне мы можем выбрать целевые функции (**Objectives**), задать варьируемый набор собственно решений задачи (**Decisions**) и сделать некоторые предположения и ограничения задачи (**Assumption**).

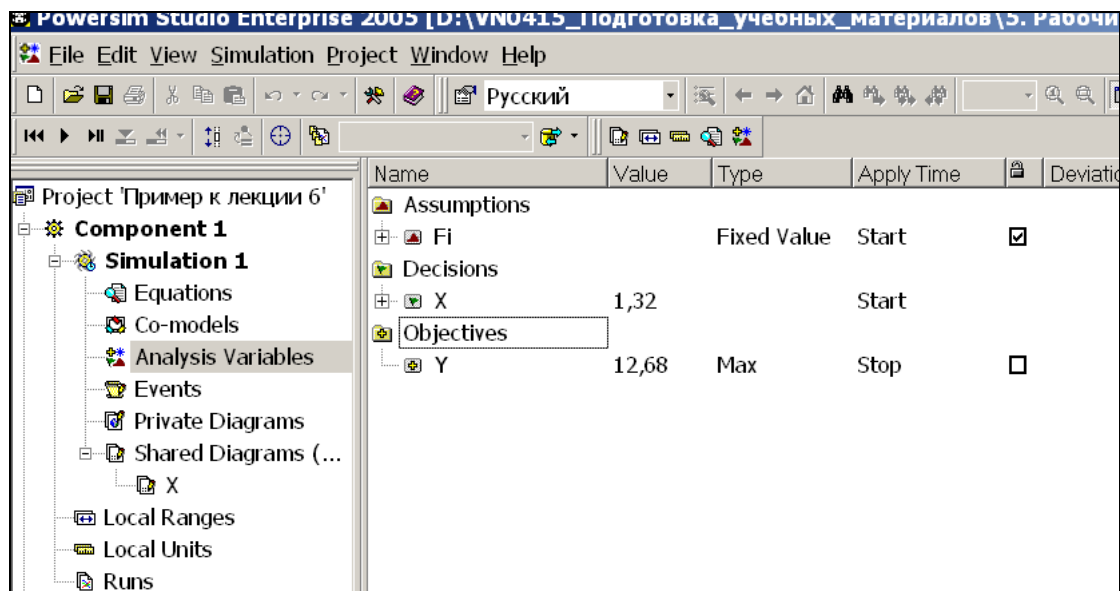


Рис. 3-3. Начало работы с оптимизатором. Окно Analysis Variables.

3.4 Выбор целевой функции Objectives.

Правой кнопкой над **Objectives** выберем **Add Objective...** и добавим интересующую нас переменную (условно **Y**). Затем войдем в окно **Y Properties** (Рис. 3-4).

Настраиваемые параметры для целевой функции:

Apply time. Определяет момент времени моделирования, при котором должно быть получено условие оптимизации. Доступен любой момент времени моделирования.

Weight. Если задана множественная оптимизация – несколько целевых функций, можно задать относительную значимость (вес) каждой функции для модели.

Normalizing divisor. Нормализация значений целевой функции. Позволяет задать область изменения функции в необходимом диапазоне.

- **Operator.** Самый важный параметр оптимизации. Позволяет выбрать цель оптимизации.
- **In** – нахождение максимального экстремума в заданном диапазоне.
- **Not In** – нахождение максимального экстремума вне заданного диапазона.
- **<** - максимальный ближайший экстремум меньше заданной величины.
- **>** - максимальный ближайший экстремум ,больший заданной величины.
- **Minimum** – глобальный минимум.
- **Maximum** – глобальный максимум.

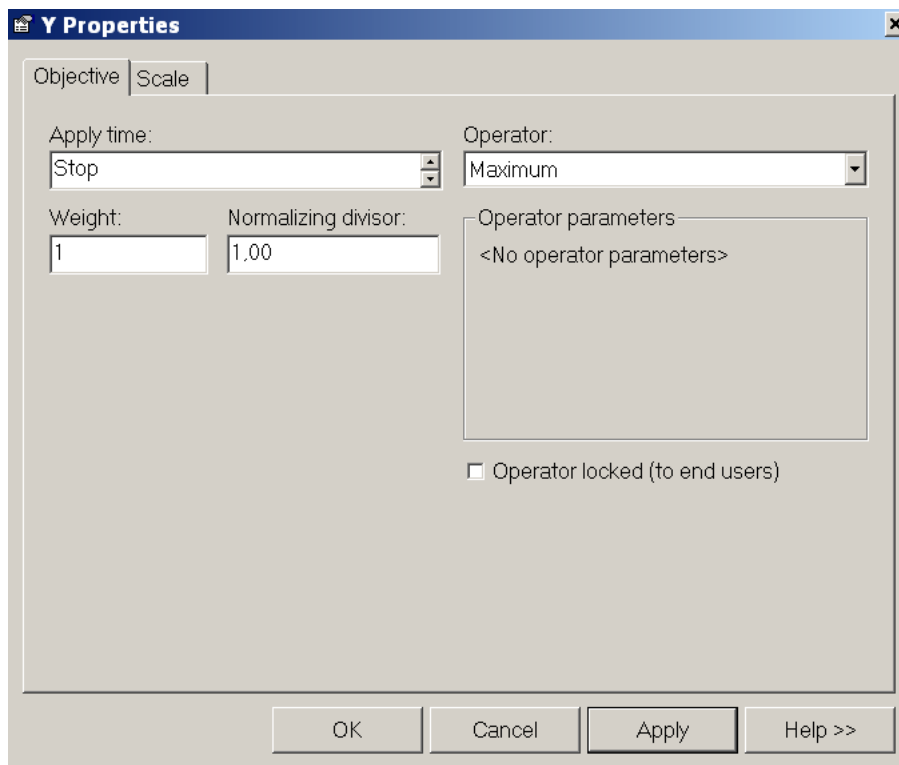


Рис. 3-4. Свойства целевой функции Objectives.

3.5 Выбор варьируемых переменных - решений задачи Decision.

Правой кнопкой выберем **Add Decision...** и добавим интересующий нас объект (условно **X**).

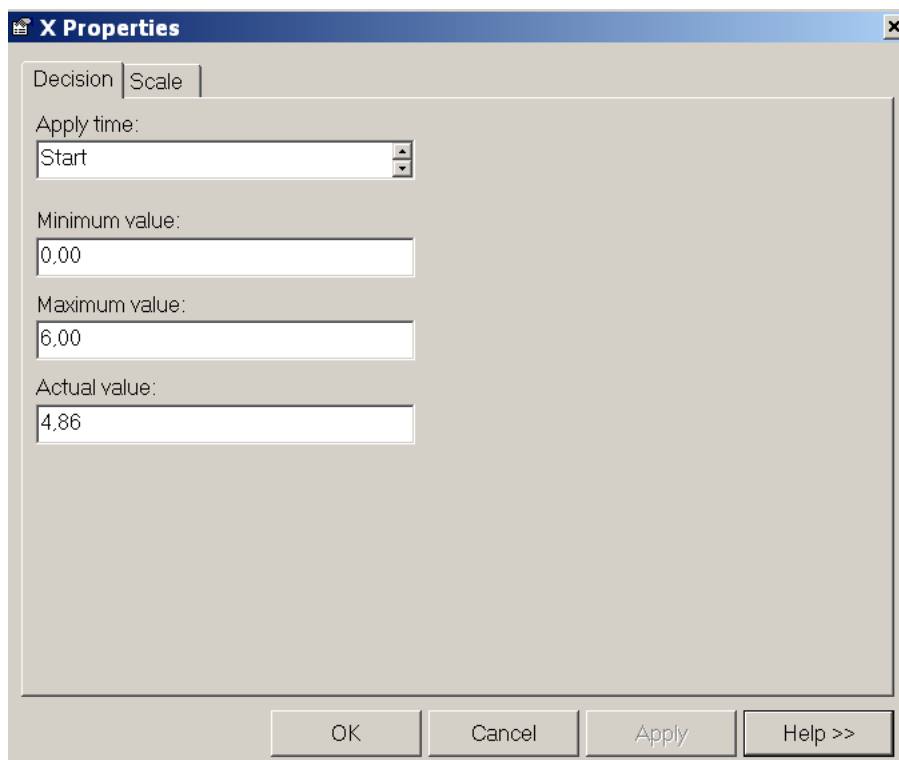


Рис. 3-5. Свойства решений Decision.

Затем войдем в окно **X Properties** (Рис. 3-5). Аналогично предыдущему пункту, можно выбрать параметры, такие как время применения и текущее значение и диапазон изменения величины.

Диапазон изменения величины решения может выступать в качестве ограничения задачи. Причем, если решение является вектором, возможно задание каждой его компоненты.

К сожалению, в эти окна не предусмотрена передача данных по ссылке, т.е. нельзя в качестве границ диапазона использовать переменную. Допускается только интерактивный ввод численных констант. Один из способов, облегчающих работу с векторами, является копирование длинной текстовой строки в окно ввода. Строка при этом набирается любым способом и не обязательно в PowerSim.

3.6 Выбор допущений Assumptions.

Правой кнопкой выберем **Add Assumption...** и добавим интересующий нас объект (условно **Fi**). Затем войдем в окно **Fi Properties** (Рис. 3-6).

Здесь возможно в рамках оптимизатора выбрать значение величины, которая не изменяется при оптимизации. Такой выбор может служить ограничением задачи. В настоящей версии доступно только Fixed value.

Все величины **Objectives**, **Decisions** и **Assumption** передаются в моделируемую задачу и запоминаются в качестве параметров до тех пор, пока они отражены в подблоке **Analysis Variables**.

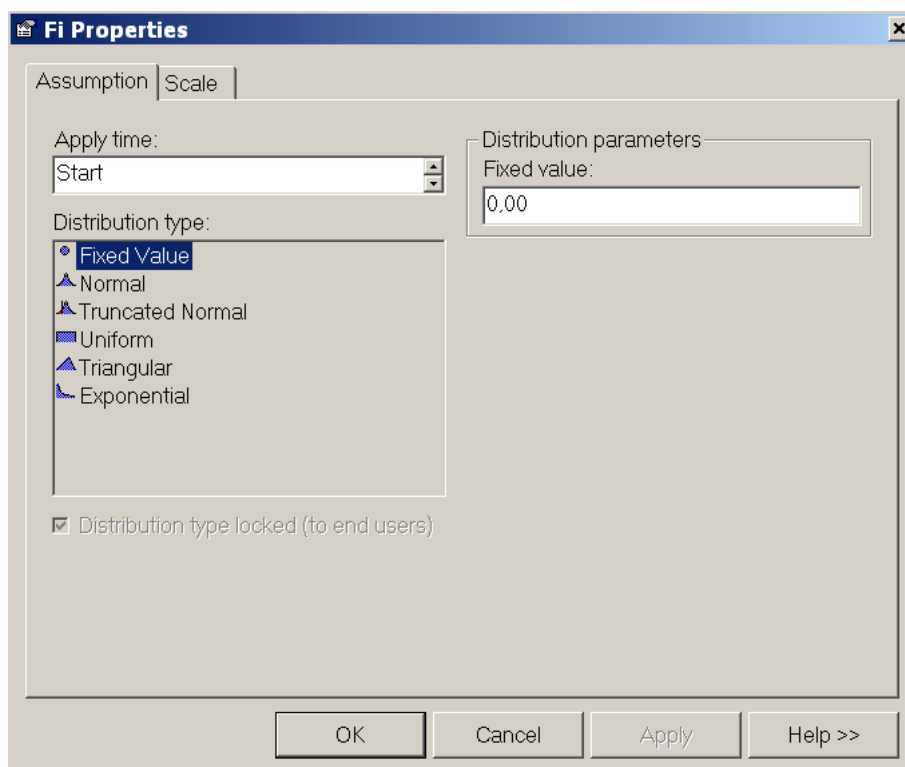



Рис. 3-6. Выбор допущений и ограничений.

3.7 Настройка параметров работы эволюционного модуля оптимизации Solver.

Нажатие на значок оптимизации  запускает оптимизатор (Рис. 3-7). Окно Solver позволяет выбрать параметры работы генетического алгоритма оптимизации.

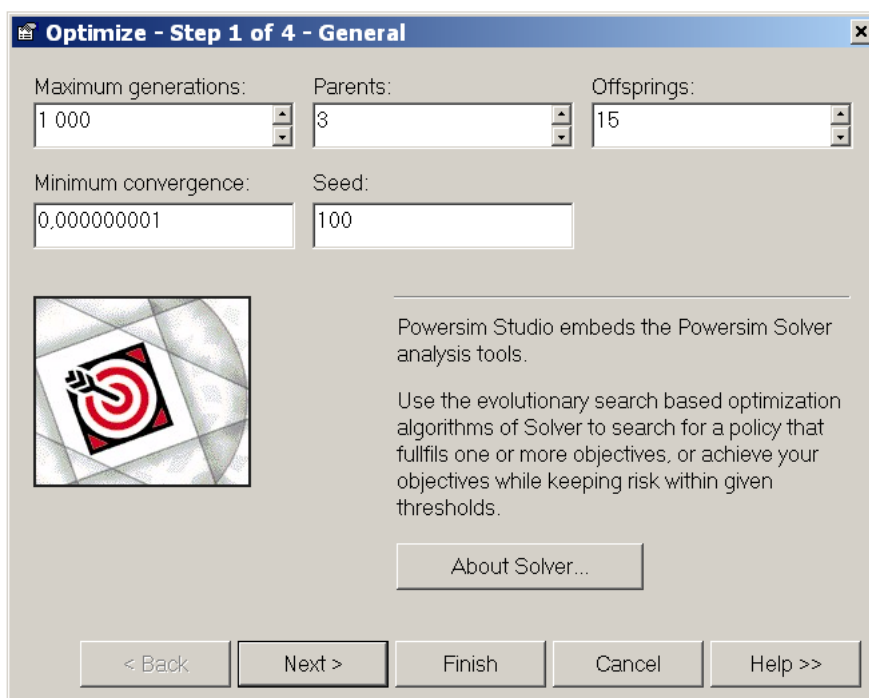


Рис. 3-7. Окно оптимизатора – Solver

Следующие операции позволяют задать параметры процесса оптимизации.

Maximum Generation -максимальное количество поколений.Под поколением здесь понимается эпоха – полный цикл решения задачи. Может лежать в диапазоне от 10 до 1000.**Parents** –число наиболее успешных родителей в текущем поколении. Диапазон изменения от 1 до 20.

Offsprings – число потомков от успешных родителей, отбираемых для следующего поколения. Может лежать в диапазоне от 5 до 100. Должно в 5 раз превышать количество родителей.

Minimum convergence - Условие остановки. Цель оптимизации считается достигнутой если последующие поколения начинают отличаться от предыдущего на некоторую заданную малую величину. Возможные значения лежат в диапазоне от 10^{-3} до 10^{-10} . **Seed** – начальная точка генератора случайных чисел. Лежит в диапазоне от 100 до 30000.

3.8 Процесс оптимизации.

Отображаются следующие параметры:

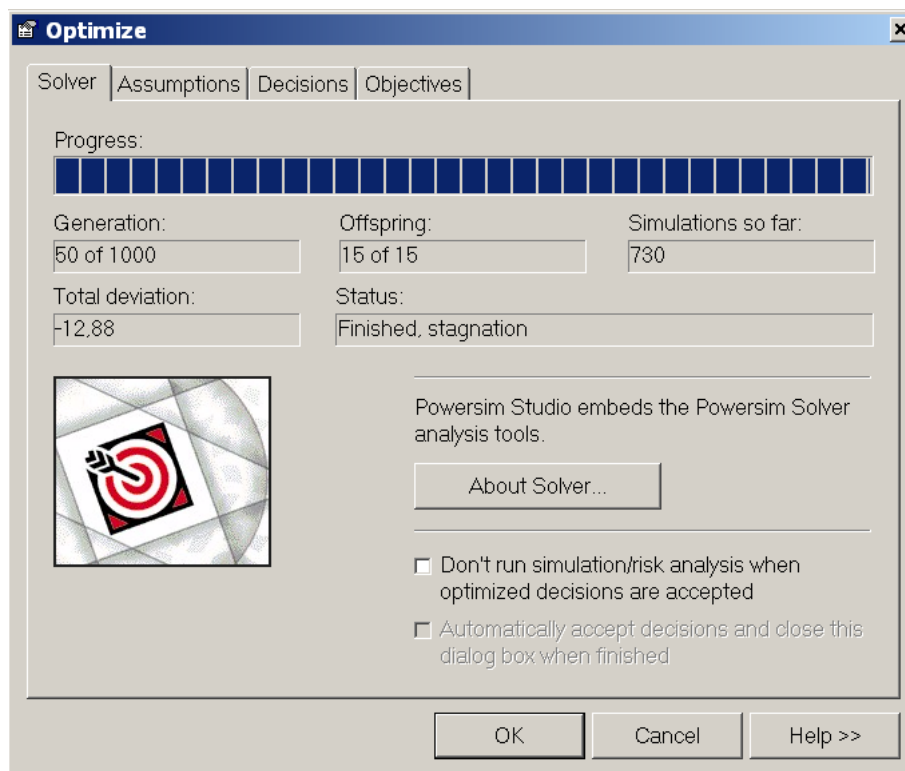


Рис. 3-8. Процесс оптимизации - Optimize

Generation – текущее поколение эволюции.

Offspring - исследуемая особь в популяции потомков.

Simulations so far – общее количество исследованных особей.

Total deviation – Общее отклонение от заданного предела. Это сумма отклонений значений каждой целевой функции, умноженной на ее вес для процесса оптимизации.

Status – статус процесса. Во время работы модуля – **Running**. Процесс оптимизации останавливается, если:

- **Finished, max generation reached.** Достигнуто максимальное (заданное) количество эпох – поколений.
- **Finished, stagnation.** Результаты в каждом новом поколении отличаются от результатов в предыдущем на величину, меньшую, чем **Minimum convergence**.
- **Running (No optimal decisions).** Оптимизатор не может найти никакого приемлемого решения.

На закладках сверху можно просмотреть значения исследуемых величин. После завершения оптимизационного процесса все данные для найденного решения передаются в исследуемую модель.

К недостаткам оптимизационного модуля можно отнести невозможность вводить ограничения в окна ввода по ссылке (по имени переменной). Решается эта проблема вводом ограничений непосредственно в модель.