

УПРАВЛЕНИЕ В СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ СИСТЕМАХ

УДК 519.688

Л.А. Демидова, Н.А. Петрова

ПРИМЕНЕНИЕ ЭВОЛЮЦИОННОГО ПОДХОДА К ЗАДАЧАМ ОПТИМИЗАЦИИ ПАРАМЕТРОВ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Рассматривается задача оптимизации параметров сложных технических систем и исследуется возможность использования эволюционного подхода для ее решения. Приведена схема работы системы оптимизации на основе эволюционного подхода. Предложен алгоритм решения задачи оптимизации на основе эволюционного подхода. Решена практическая задача оптимизации покрытия объекта наблюдения системой съемки.

Ключевые слова: *техническая система, оптимизация, параметр, эволюционный подход, эволюционный алгоритм.*

Введение. Целью работы является применение эволюционного подхода к задачам оптимизации параметров сложных технических систем посредством разработки программного инструмента, интегрирующего в себе математические модели и поисковые методы эволюционных алгоритмов и обеспечивающего поиск оптимальных значений параметров задачи оптимизации, удовлетворяющих большому количеству требований и ограничений, с приемлемыми временными затратами.

В данном случае под задачами оптимизации параметров сложных технических систем подразумевается класс многопараметрических однокритериальных задач оптимизации подбора значений параметров, позволяющих достигнуть максимально возможной эффективности функционирования системы.

Оптимизация параметров сложных технических систем. Оптимизация параметров сложной технической системы заключается в нахождении значений параметров системы, соответствующих оптимальному по определенным критериям режиму ее работы [8].

Решение таких задач оптимизации осложняется тем, что начальные значения параметров, от которых следует двигаться для нахождения оптимального набора значений, не заданы. При этом известны только возможные или допустимые диапазоны изменения значений параметров,

имеется некоторый способ оценки качества функционирования системы на том или ином наборе значений параметров, и, кроме того, могут быть заданы те или иные дополнительные ограничения, накладываемые предметной областью решаемой задачи. В этом случае решением задачи оптимизации должен являться удовлетворяющий всем ограничениям набор значений параметров, соответствующий максимальному качеству работы системы [8].

Для задач оптимизации параметров сложных технических систем использование эволюционного подхода, представляющего перспективное и динамично развивающееся направление интеллектуальной обработки данных, обеспечивает возможность получения искомого решения за приемлемое время при наличии минимального объема исходных данных [8].

Эволюционный подход к задачам оптимизации. Эволюционный подход к предметным задачам оптимизации позволяет применять в качестве методов оптимизации эволюционные алгоритмы (генетические алгоритмы, эволюционные стратегии, эволюционное программирование), реализующие возможность одновременного поиска среди нескольких альтернативных вариантов решений и выбора лучших из них [1, 2, 4, 6].

Можно выделить четыре основных аспекта, рассматриваемых при решении практических

задач с использованием эволюционных алгоритмов [5]:

- выбор способа представления (кодирования) решения;
- выбор (разработка) эволюционных операторов;
- определение законов выживания решений;
- создание начальной популяции решений.

Для представления (кодирования) решения в виде, удобном для реализации эволюционных алгоритмов, требуется определение такой структуры, которая позволит кодировать любое возможное решение и производить его оценку. Потенциальный вариант представления решения должен обеспечивать возможность проведения различных перестановок в наборах параметров оптимизации [3].

В качестве эволюционных операторов могут использоваться любые известные операторы скрещивания, мутации и т.д. Кроме того, возможно дополнение старых и создание новых стратегий случайных изменений анализируемых решений с учетом особенностей предметной области задачи оптимизации [3].

При реализации эволюционных алгоритмов могут применяться различные законы выживания решения, однако наиболее часто реализуется закон выживания сильнейших, обеспечивающий выбор решений с наилучшими значениями функций приспособленности [3].

Создание начальной популяции возможно только с применением знаний о предметной области задачи оптимизации, поскольку для адекватного применения эволюционного алгоритма, одной из основных особенностей которого является представление любого альтернативного решения в виде строки параметров, зависящих (явно или неявно) от аргументов функции приспособленности, необходимо правильно представить реальные оптимизируемые параметры, найденные значения которых должны обеспечить оптимальное значение функции приспособленности.

Универсальность эволюционного подхода заключается в том, что от конкретной задачи оптимизации зависят только критерии поиска (в частном случае, если критерий один, то в качестве него используется функция приспособленности эволюционного алгоритма) и способ представления (кодирования) решений. Все остальные действия при реализации эволюционных алгоритмов для задач оптимизации производятся аналогично, но с применением различных эволюционных операторов [3].

При применении эволюционного подхода можно задачу оптимизации представить в виде

модели предметной области, интегрированной в эволюционные алгоритмы [6].

В данном случае под моделью предметной области понимается представление набора параметров задачи, а также критерия ее оптимизации в приемлемом для использования эволюционного алгоритма виде.

Система оптимизации на основе эволюционного алгоритма. Основными входными параметрами системы оптимизации на основе эволюционного алгоритма являются: количество параметров оптимизации с диапазонами их возможных изменений, функция приспособленности, размер популяции, количество популяций, начальные значения вероятностей скрещивания и мутации, количество итераций эволюционного алгоритма для расчета функции приспособленности.

Эволюционный подход к задаче оптимизации реализован с использованием эволюционных стратегий, являющихся, как и генетические алгоритмы, разновидностью эволюционных алгоритмов и представляющих собой эвристические методы оптимизации, основанные на адаптации и эволюции [2, 4].

Как любые эволюционные алгоритмы, эволюционные стратегии предполагают формирование исходной популяции хромосом, которая подвергается селекции и воздействию эволюционных операторов с целью получения наиболее приспособленных хромосом.

Главные отличия эволюционных стратегий от других эволюционных алгоритмов заключаются в следующем [7].

1. Эволюционные стратегии используют представление хромосом в виде векторов действительных чисел, соответствующих значениям параметров задачи оптимизации.

2. При реализации эволюционной стратегии процедура селекции хромосом и формирования новой популяции предполагает создание промежуточной популяции, состоящей из всех хромосом-родителей и хромосом-потомков, полученных в результате применения эволюционных операторов, с последующим уменьшением размера промежуточной популяции до размера родительской популяции за счет исключения наименее приспособленных хромосом.

3. При реализации эволюционных стратегий сначала производится процедура рекомбинации хромосом (посредством скрещивания и мутации хромосом-родителей), а затем – процедура селекции, в то время как в других эволюционных алгоритмах используется обратный порядок этих процедур.

4. При реализации эволюционных стратегий

возможна самоадаптация таких параметров, как вероятности скрещивания и мутации. Так, если при реализации эволюционных стратегий на некоторой итерации потомок не удовлетворяет всем ограничениям, то производится увеличение значения счетчика «плохо приспособленных» хромосом, но, тем не менее, «плохо приспособленный» потомок включается в новую промежуточную популяцию. Если таких «плохо приспособленных» потомков в популяции оказывается много, и их количество превышает некоторое пороговое значение, то осуществляется запуск процесса адаптации параметров, например, реализуется увеличение вероятности скрещивания.

За счет перечисленных отличий эволюционные стратегии имеют малую временную сложность и обладают возможностью адаптации параметров эволюционного алгоритма [9], причем альтернативные решения представляются единым массивом числовых значений оптимизируемых параметров задачи, а воздействие на данные массива осуществляется с учетом их смыслового содержания и направлено на улучшение значений оптимизируемых параметров по некоторому заданному критерию, что позволяет находить более эффективные решения сложных задач за меньшее время [6].

Схема работы системы оптимизации на основе эволюционного алгоритма представлена на рисунке 1.

Связи блоков на схеме реализуют следующие функции [3]:

- связь 1 – поиск решения на основе эволюционного алгоритма;
- связь 2 – представление данных предметной области и передача оптимизируемых значений параметров, полученных с помощью блока оптимизации;
- связь 3 – определение стратегии поиска эволюционного алгоритма;
- связь 4 – передача полученных эволюционным алгоритмом значений оптимизируемых параметров в базу данных.

Программная система оптимизации на основе эволюционного алгоритма обеспечивает:

- поиск наборов значений параметров задачи оптимизации в соответствии с моделью предметной области;
- определение «лучшего» набора значений параметров задачи оптимизации среди всей совокупности, полученных за определенное количество итераций решений;
- работу с базой данных по извлечению исходной информации и передаче полученных данных.



Рисунок 1 – Схема работы системы оптимизации на основе эволюционного алгоритма

Эволюционный алгоритм решения задачи оптимизации. Использование эволюционного алгоритма позволяет свести решение задачи оптимизации к выполнению следующих двух шагов.

Шаг 1. Формализация задачи оптимизации.

На данном шаге осуществляется задание параметров оптимизации, диапазонов их допустимых значений, критерия оптимизации (функции приспособленности) и вспомогательных параметров (например, констант, используемых при расчетах).

Шаг 2. Оптимизация и анализ результатов оптимизации.

На данном шаге реализуется запуск эволюционного алгоритма оптимизации, выявление наилучших значений параметров задачи оптимизации для текущей итерации эволюционного алгоритма и их представление в удобном для анализа виде.

Эволюционный алгоритм решения задачи оптимизации представлен на рисунке 2.

Блок алгоритма «Задание параметров и ограничений задачи оптимизации» представляет этап постановки задачи, на котором необходимо:

- задать параметры, изменением которых предполагается обеспечить максимальную эффективность решения задачи оптимизации в смысле достижения оптимального значения используемого критерия (функции приспособленности);
- задать критерий (функцию приспособленности), что позволит отразить взаимосвязь между параметрами задачи и обеспечить выполнение сравнения различных вариантов полученных решений для определения «лучшего»;
- задать диапазоны изменения оптимизируемых параметров задачи;

– определить предельные значения параметров задачи с помощью установки ограничений («равенство» или «неравенство»).

Блок алгоритма «Реализация эволюционного алгоритма для заданных параметров» отвечает за реализацию эволюционного алгоритма оптимизации, в котором генерируются наборы из значений варьируемых параметров и определяются хромосомы после их скрещивания и мутации за заданное количество итераций.



Рисунок 2 – Эволюционный алгоритм решения задачи оптимизации

Блок алгоритма «Расчет функции приспособленности и упорядочивание результатов» и блок «Отбор лучших по функции приспособленности хромосом (наборов значений параметров) для последующей итерации» осуществляют рас-

чет функции приспособленности, что позволяет сравнить и выбрать «лучшие» наборы значений оптимизируемых параметров и использовать их для дальнейших итераций эволюционного алгоритма. Кроме того, здесь производится проверка на соблюдение всех ограничений предметной области, налагаемых на значения варьируемых параметров и функцию приспособленности.

Блок алгоритма «Отображение результатов» отвечает за вывод списка значений функции приспособленности (обеспечивающей достижение цели критерия оптимизации) и соответствующих им наборов значений оптимизируемых параметров и значений ограничений.

Экспериментальные исследования. Эволюционный алгоритм оптимизации покрытия объекта наблюдения съемкой. При моделировании функционирования сложных технических систем важной задачей является определение таких оптимальных значений параметров, которые позволят достигнуть максимальной эффективности работы системы [8].

Для проверки работоспособности эволюционного алгоритма и программной системы оптимизации была рассмотрена практическая задача оптимизации покрытия объекта наблюдения системой съемки.

При проведении экспериментальных исследований предварительно было произведено планирование съемки объектов наблюдения для выбранного района с целью получения начальных значений параметров съемки объектов наблюдения. В таблице 1 приведены начальные значения оптимизируемых параметров съемки и соответствующие интервалы допустимых изменений для одного из объектов съемки.

Границы съемки, рассчитанной по начальным значениям параметров, схематично представлены на рисунке 3 (темный прямоугольник – объект; светлый многоугольник – захват съемки).

Критерием оптимального покрытия объекта наблюдения является минимальное значение разности площади покрытия и площади объекта.

При решении задачи оптимизации покрытия объекта системой съемки необходимо подобрать значения параметров съемки, обеспечивающие оптимальное покрытие объекта наблюдения. Для решения данной задачи оптимизации применим описанный выше эволюционный алгоритм.

При этом предварительно необходимо реализовать представление предметной области задачи оптимизации в терминах эволюционного алгоритма. Искомое решение задачи оптимизации покрытия объекта наблюдения съемкой может быть представлено в виде хромосомы, в ко-

торой закодированы параметры задачи, такие как широта и долгота точки начала съемки, время включения съемки, длительность съемки, скорость движения изображения и азимут, которые являются точками в пространстве поиска.

Таблица 1 – Начальные значения параметров съемки и интервалы допустимых значений

Параметр съемки	Начальное значение	Интервал допустимых значений
Широта точки начала съемки, град.	62,5856	(61; 62,6)
Долгота точки начала съемки, град.	- 163,171	(- 165; -162)
Время включения съемки, сек	7888174	(7888150; 7888200)
Длительность съемки	11,8483	(5; 15)
Скорость движения изображения	60	(45; 75)
Азимут	0	(0; 360)

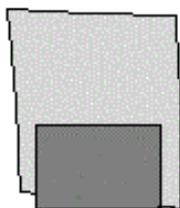


Рисунок 3 – Границы съемки, рассчитанной по начальным значениям параметров

Тогда популяция решений размером n может быть записана как:

$$P_1 = (nshir_1, ndolg_1, ontel_1, dlit_1, sdi_1, azim_1), \quad (1)$$

$$P_n = (nshir_n, ndolg_n, ontel_n, dlit_n, sdi_n, azim_n),$$

где P_i – i -я хромосома ($i = \overline{1, n}$); $nshir_i$ – широта точки начала съемки; $ndolg_i$ – долгота точки начала съемки; $ontel_i$ – время включения съемки; $dlit_i$ – длительность съемки; sdi_i – скорость движения изображения съемки; $azim_i$ – азимут; n – размер популяции.

Для каждого гена в хромосоме P_i ($i = \overline{1, n}$) в соответствии с ограничениями, накладываемыми предметной областью задачи оптимизации, определен интервал допустимых значений:

- $61 < nshir_i < 62,6$;
- $-165 < ndolg_i < -162$;
- $7888150 < ontel_i < 7888200$;
- $5 < dlit_i < 15$;
- $45 < sdi_i < 75$;
- $0 < azim_i < 360$.

Если в качестве генов (атомарных элементов генотипа) рассматривать широту точки начала съемки, долготу точки начала съемки, время включения съемки, длительность съемки, скорость движения изображения и азимут, то фенотипом будет являться соответствующий генотипу тот или иной набор (вариант) значений параметров съемки.

Тогда задачей эволюционного алгоритма при оптимизации покрытия объекта наблюдения системой съемки является поиск оптимальных значений параметров съемки: $nshir$, $ndolg$, $ontel$, $dlit$, sdi , $azim$.

При решении задачи оптимизации покрытия объекта наблюдения системой съемки с использованием эволюционного алгоритма целесообразно использовать:

- несколько популяций фиксированного размера;
- фиксированный размер популяции, соответствующий некоторому количеству вариантов решения задачи оптимизации, представляющих собой наборы значений параметров съемки;
- фиксированную длину (разрядность) хромосом, равную шести и соответствующую количеству оптимизируемых параметров;
- одинаковые комбинации стратегий отбора и формирования следующего поколения в каждой популяции;

– случайный отбор, предполагающий, что при выполнении операции скрещивания хромосомы определяющие наборы значений параметров съемки выбираются случайным образом, поскольку изначально популяции хромосом сформированы из наборов неоптимальных значений параметров съемки;

– одноточечный кроссовер (скрещивание) и одноточечную мутацию, что связано с содержанием в хромосомах различных по смыслу параметров.

Каждая новая популяция хромосом размером n текущей итерации эволюционного алгоритма формируется из промежуточной популяции хромосом, состоящей из хромосом-родителей и хромосом-потомков, посредством удаления наименее приспособленных хромосом (с большими значениями функций приспособленности) так, чтобы поддерживать фиксированный размер популяции и обеспечивать возможность замены хромосом в популяции на «лучшие», стремящиеся к оптимальным значениям параметров задачи оптимизации.

При реализации скрещивания в эволюционном алгоритме сначала осуществляется случайный выбор родителей – двух хромосом, затем случайным образом выбирается точка скрещи-

вания и, наконец, реализуется скрещивание (обмен частями) хромосом-родителей и получение двух хромосом-потомков.

При реализации мутации в эволюционном алгоритме сначала для некоторой хромосомы-родителя случайным образом выбирается точка мутации, а затем реализуется мутация и получение хромосомы-потомка.

Периодически (например, через установленное количество итераций эволюционного алгоритма) производится случайный обмен хромосомами между популяциями, что позволяет реализовать разновидность параллельного эволюционного алгоритма, обладающего некоторыми свойствами островной модели генетического алгоритма [2, 4].

Островная модель подразумевает, что есть группа «близкорасположенных островов», на которых эволюционируют однотипные популяции. Эти популяции развиваются независимо, и только изредка происходит обмен хромосомами между ними.

Островная модель предполагает наличие нескольких популяций одинакового фиксированного размера; фиксированную разрядность хромосом; возможность использования любых комбинаций стратегий отбора и формирования поколений в популяциях; отсутствие ограничений, накладываемых на операции скрещивания и мутации; случайный обмен хромосомами между популяциями.

Анализ предметной задачи оптимизации покрытия объекта наблюдения системой съемки показывает, что изначально формула для функции приспособленности может быть записана как:

$$S_s - S_{ob} \rightarrow \min, \quad (2)$$

где S_s – площадь съемки, S_{ob} – площадь объекта.

С учетом того, что необходимо обеспечить максимальное приближение площади съемки к площади объекта наблюдения так, чтобы вершины объекта наблюдения находились внутри границ оптимальной съемки, а площадь съемки отличалась от площади объекта наблюдения на минимальную величину, уточненная формула для функции приспособленности приобретает вид:

$$(S_s - S_{ob}) + N \cdot N_{kr} \rightarrow \min, \quad (3)$$

где S_s – площадь съемки; S_{ob} – площадь объекта; N – количество вершин объекта, не попавших внутрь границ съемки; N_{kr} – числовая величина, много большая разности площадей и необходимая для достижения значительного ухудшения функции приспособленности в случае несо-

блюдения условия попадания всех вершин объекта в площадь съемки.

Исходя из того, что разность площадей съемки и объекта будет стремиться к минимальному значению квадратных километров, можно принять значение величины N_{kr} равным 100000, что позволит увеличить разность площадей в сотни раз и обеспечит отбраковку решений, в которых не все вершины объекта охвачены.

Так как площадь съемки должна покрывать площадь объекта, то необходимо обеспечить выполнение условия положительности для функции приспособленности:

$$((S_s - S_{ob}) + N \cdot N_{kr}) > 0. \quad (4)$$

Эволюционный алгоритм оптимизации покрытия объекта наблюдения съемкой может быть представлен следующей последовательностью действий.

1. Создание M начальных популяций размером n из хромосом P_i ($i = \overline{1, n}$).

2. Если текущее количество g итераций эволюционного алгоритма меньше максимального количества итераций G , случайным образом происходит выбор хромосом-родителей из популяций и осуществляется переход к шагу 3. При достижении максимального количества итераций G выполняется переход к шагу 6.

3. Если текущее количество итераций g некратно некоторому числу I ($I < G$), то операции скрещивания и мутации реализуются последовательно в каждой популяции. Если текущее количество итераций g кратно числу I , то из M популяций случайным образом выбираются T ($T \leq M/2$) пар популяций, в каждой такой паре популяций формируется пара хромосом, лучших по функции приспособленности в своей популяции, к которым затем применяется операция скрещивания. В данном случае I – количество итераций, позволяющее ускорить время выполнения эволюционного алгоритма путем перераспределения генов в хромосоме при скрещивании хромосом из разных популяций через каждые I итераций.

4. Создание промежуточной популяции из родителей и потомков, вычисление функции приспособленности по формуле (3) и проверка ограничения (4) для каждой хромосомы.

5. Создание новой популяции размером n путем исключения наименее приспособленных хромосом. Переход к шагу 2.

6. Выбор среди популяций «лучшей» хромосомы, имеющей минимальное значение функции приспособленности вида (3), определяющей на-

бор значений параметров. Вывод полученного набора оптимизируемых значений параметров.



Рисунок 4 – Границы съемки, рассчитанной по оптимальным значениям параметров

После завершения работы эволюционного алгоритма рекомендуется либо воспользоваться полученным с помощью эволюционного алгоритма решением – набором найденных значений параметров съемки, либо увеличить число итераций эволюционного алгоритма для дальнейшего поиска.

Оптимизация параметров

Исходные данные

Параметры	Наименование	Начальное значение	Ограничение ">"	Ограничение "<"
Параметр 1	Широта точки начала съемки	62,5856	61	62,6
Параметр 2	Долгота точки начала съемки	-163,171	-165	-162
Параметр 3	Время включения съемки	7888174	7888150	7888200
Параметр 4	Длительность съемки	11,8483	5	15
Параметр 5	Скорость движения изображения	60	45	75
Параметр 6	Азимут	0	0	360

Результаты расчетов

Итерация	Широта точки	Долгота точки	Время включения	Длительность	Скорость	Азимут	ФП
700	62,328692	-163,218283	7888152,891757	11,812061	45,000000	0,380495	6624,782
750	62,386961	-163,199392	7888154,313636	10,723838	45,000000	1,631957	5341,250
800	62,431231	-163,129544	7888156,991453	9,739206	45,000000	1,490934	4559,977
850	62,460782	-163,145325	7888155,489741	8,904615	45,000000	1,086995	3261,092
900	62,495270	-163,161418	7888154,266380	8,495759	45,180437	1,006001	2699,069
950	62,512768	-163,183171	7888150,000000	8,251402	45,000000	0,675263	2043,278
1000	62,514737	-163,187060	7888150,000682	8,070748	45,000000	0,706152	1888,631

Рисунок 5 – Диалоговое окно программной системы оптимизации на основе эволюционного алгоритма, содержащее наборы значений оптимизируемых параметров

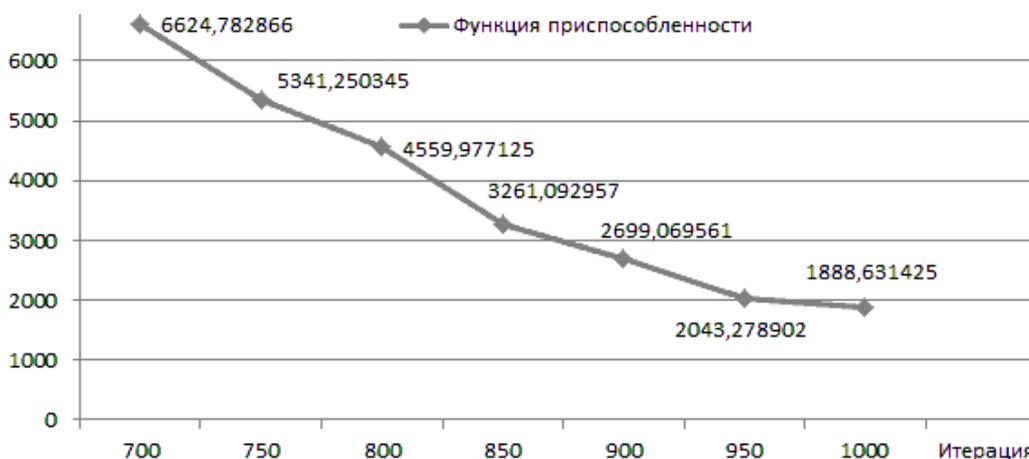


Рисунок 6 – График зависимости функции приспособленности от количества итераций эволюционного алгоритма

Оптимизация покрытия объекта наблюдения при съемке. С использованием предло-

женного эволюционного алгоритма был выполнен расчет оптимальных параметров съемки по-

крытия объекта наблюдения при заданных начальных значениях параметров (таблица 1), ограничениях, накладываемых на значения параметров (таблица 1), количестве итераций эволюционного алгоритма – 1000, погрешности достижения минимума функции приспособленности – 0,0001, количестве популяций – 2, количестве потомков в операторах скрещивания – 2, размере популяции – 20.

Границы съемки, рассчитанной по полученным значениям параметров съемки, схематично представлены на рисунке 4 (темный прямоугольник – объект; светлый многоугольник – захват съемки).

На рисунке 5 показано диалоговое окно программной системы оптимизации на основе эволюционного алгоритма, содержащее наборы значений оптимизируемых параметров. На рисунке 6 приведен график зависимости функции приспособленности от количества итераций эволюционного алгоритма, отражающий улучшение функции приспособленности при увеличении количества итераций в соответствии с представленными на рисунке 5 результатами расчетов.

Полученные результаты свидетельствуют о целесообразности применения эволюционного алгоритма для решения задачи оптимизации покрытия объекта наблюдения системой съемки, подразумевающей, что расхождение между площадью объекта и площадью съемки должно быть минимальным и площадь съемки должна покрывать площадь объекта при заданном количестве итераций алгоритма.

Заключение. Разработанная программная система оптимизации на основе эволюционного алгоритма является инструментом поиска оптимальных значений параметров сложных технических систем, благодаря реализованной в ней возможности представления оптимизационной задачи в виде модели предметной области, интегрированной в эволюционные алгоритмы.

Рассмотренный практический пример решения задачи оптимизации покрытия объекта наблюдения системой съемки наглядно продемонстрировал возможности предложенного программного инструмента при решении прикладной оптимизационной задачи.

Усовершенствование эволюционного алгоритма, примененного к решению покрытия объекта наблюдения системой съемки, может быть выполнено в плане модернизации стратегии и способа генерации популяций и потомков, а также подбора оптимального количества популяций и количества потомков.

Целью дальнейших исследований по применению эволюционного подхода к задачам оптимизации параметров сложных технических систем является разработка модели предметной области задачи оптимального планирования съемки группы объектов с приемлемыми временными затратами при наличии тех или иных критериев оптимизации съемки (захвата большего количества объектов, минимизации затрат съемки и т.п.).

Библиографический список

1. *Афонин П.В.* Система оптимизации на основе имитационного моделирования, генетического алгоритма и нейросетевых метамоделей // Proceedings of the XIII-th International Conference «Knowledge-Dialogue-Solutions». – Varna, Bulgaria, 2007. – V.1. – P.60-63.
2. *Демидова Л.А., Кираковский В.В.* Методы кластеризации объектов на основе нечетких множеств второго типа и генетического алгоритма // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление. – СПб., 2008. – №6(69). – С. 136-142.
3. *Демидова Л.А., Петрова Н.А.* Программная система оптимизации на основе эволюционного подхода // Информационные технологии в учебном процессе и научных исследованиях: межвузовский сборник научных трудов молодых ученых, специалистов и студентов / под ред. А.Н. Пылькина. – Рязань: РГРТУ. – 2012. – С. 45-47.
4. *Демидова Л.А., Скворцова Т.С.* Применение генетических алгоритмов для прогнозирования нечетких временных рядов // Вестник Рязан. гос. радиотехн. ун-та. – 2010. – № 1 (вып. 31). – С. 95-98.
5. *Емельянов В.В.* и др. Теория и практика эволюционного моделирования. – М.: ФИЗМАТЛИТ, 2003. – 432 с.
6. *Петрова Н.А.* Решение задач оптимизации методами эволюционных вычислений // Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: 17-я международная научно-техническая конференция, посвященная 60-летию РГРТУ. – Рязань: РГРТУ. – 2012. – С. 113-114.
7. *Рутковская Д., Пилиньский М., Рутковский Л.* Нейронные сети, генетические алгоритмы и нечеткие системы: пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2006. – 452 с.: ил.
8. *Токарева Н.А.* Применение эволюционных технологий при оптимизации параметров сложных технических систем // Математическое и программное обеспечение вычислительных систем: межвузовский сборник научных трудов / под ред. А.Н. Пылькина. – Рязань: РГРТУ. – 2011. – С. 46-48.
9. *Токарева Н.А.* Сущность и применение эволюционных стратегий в задачах оптимизации // Новые информационные технологии в научных исследованиях и в образовании. НИТ-2010: XV Всероссийская научно-техническая конференция студентов, молодых ученых и специалистов. – Рязань: РГРТУ. – 2010. – С. 101-103.

УДК 621.317.75:519.2

В.В. Белов, М. В. Наумович**СТРУКТУРЫ ДАННЫХ
И АЛГОРИТМЫ ПЛАНИРОВАНИЯ СОБЫТИЙ РАБОТ
С НЕОПРЕДЕЛЕННЫМИ ВРЕМЕННЫМИ ХАРАКТЕРИСТИКАМИ**

Предложены вариант реализации структуры представления данных о работах с неопределенными временными характеристиками, а также алгоритмы динамического планирования с учетом работ такого рода.

Ключевые слова: алгоритм, данные, работа, планирование, неопределенность, онтология.

Введение. Проблема эффективного планирования времени в той или иной степени стоит перед каждым человеком. На текущий день для этого существует специальная концепция тайм менеджмента – технологии организации времени и повышения эффективности целенаправленной деятельности [1, 2]. Одной из основных трудностей при планировании времени является вид работ, у которых нет четких рамок. Такие дела могут накапливаться продолжительное время в больших количествах и приводить к стрессовым ситуациям. Человеку приходится периодически просматривать длинный список дел, которые он не смог сразу привязать к определенному времени, чтобы выполнить их в свободное время. Это также требует сбора и систематизации всех работ подобного рода. Сегодня одним из главных инструментов планирования являются специальные программы для ЭВМ, однако они практически не содержат в себе интеллектуальной составляющей для автоматизированного построения плана действий пользователя [3].

Цель данной работы – разработка алгоритмов и структур данных для обеспечения стратегии планирования работ с неопределенными временными характеристиками в программах для ЭВМ.

Представление данных о работах. Первым шагом к реализации интеллектуальной программы планировщика является проектирование системы представления знаний. Анализ опыта представления знаний, связанных с пространственно-временными отношениями [4, 5], показывает, что наиболее адекватным для этой цели является использование механизма онтологий.

Для моделирования онтологии «умного» планировщика была использована среда Protégé 3.4.4. Схема классов онтологии представлена на рисунке 1. Прежде всего, в онтологии присутствуют традиционные для данной предметной области сущности.

Работа – безусловно, такая сущность есть в концепции любого современного планировщика. Однако в совокупности с другими элементами разработанной онтологии она представляет собой гораздо более сложную структуру. В частности, обладает расширенным набором условий, а также специальными сущностями для повторения и напоминания. В приведенной онтологии все типы работ представляются одной универсальной сущностью, в то время как в большинстве планировщиков для этой цели используются несколько.

Список работ – базовая сущность для представления различных наборов работ.

План – содержит набор работ, которые пользователь собирается выполнить. В данной сущности объединены все виды предстоящих задач независимо от их временных характеристик.

Условие – базовая сущность для всех представленных в онтологии классов условий.

Приоритет – содержит информацию о важности работ для пользователя. Обычно берет на себя лишь информативную роль, но в предложенной концепции является важным параметром для автоматизации построения плана действий. При отсутствии четких временных условий приоритет становится первостепенным фактором, от которого зависит очередность включения той или иной работы в предлагаемый пользователю список дел.

Тег – базовая сущность для всех классов информативных меток, которые можно присваивать работам.

Категория и проект – две основных сущности меток, призванных объединять работы по их фактическому и общесмысловому содержанию. При отсутствии или неопределенности условий могут служить единственным средством для автоматизации построения плана действий.

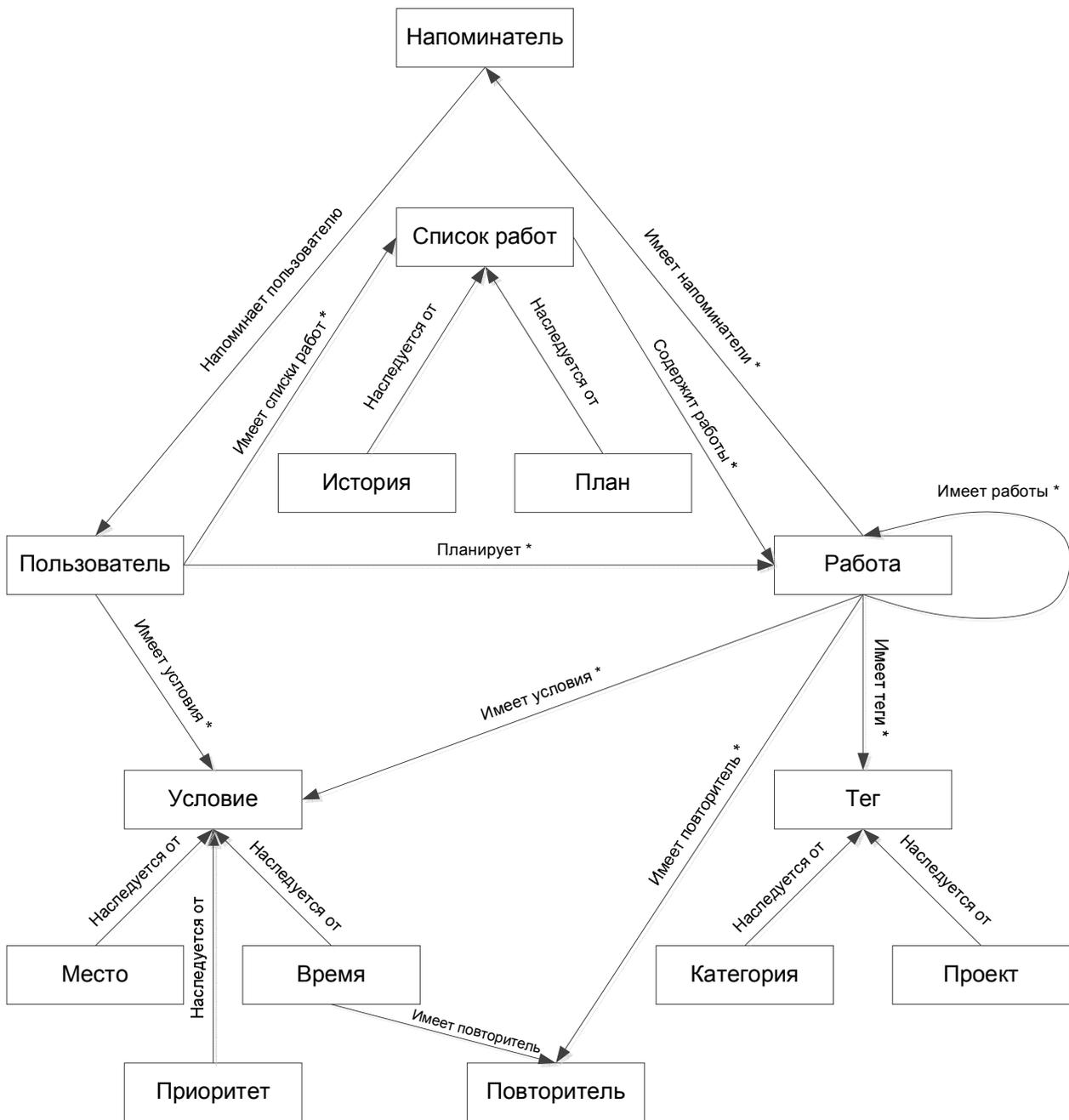


Рисунок 1 – К-диаграмма онтологии интеллектуальной программы планировщика

Непосредственно для решения поставленных ранее проблем, наличествующих в современных средствах планирования, были введены следующие классы.

Место – представление сведений о местоположении субъекта или объекта. Фактор местоположения объектов играет в жизни человека одну из главных ролей. Наряду со временем, пространство составляет единую систему координат человечества, относительно которой могут быть описаны любой объект или процесс. Сегодня в планировщиках местоположение объектов представлено весьма условно. Чаще всего пользователь может ввести условное описание места планируемой работы, например «г. Рязань,

ул. Ленина, д. 1» или просто «Офис», реже можно отметить точку на географической карте сервисов вроде «Карты Google» или «Яндекс. Карты». Эта информация, безусловно, может быть полезна пользователю, однако для планировщика является неинформативной. Для повышения эффективности взаимодействия пользователя с планировщиком предлагается использовать географические координаты мест работ и расположения пользователя как одни из основных данных для автоматизации построения плана действий. Таким образом, положение человека относительно мест предстоящих работ будет давать планировщику важную дополнительную информацию.

Время – содержит сведения о временной ориентации работ. Новая организации этой традиционной составляющей концепции представления работ в планировщике включает в себя три компонента: начало действия, конец действия и длительность действия, причем неопределенность одной из компонент также является полноправной составляющей временных параметров. Это позволяет добиться представления новых типов работ с неопределенными временными характеристиками с возможностью последующей автоматизированной обработки, не реализованной в современных планировщиках.

История – представление сведений о ранее выполненных пользователем работах. На основе данной базы уже случившихся фактов алгоритмы интеллектуального планирования могут вырабатывать пути построения нового плана действий в условиях неопределенности временных характеристик грядущих работ.

Пользователь – содержит модель субъекта, использующего планировщик. Служит для хранения данных, необходимых для организации интерактивного взаимодействия между пользователем и программой. В частности, класс хранит связанные данные о местоположении пользователя и истории выполненных заданий. Эти данные необходимы для организации автоматического планирования работ с неопределенными временными характеристиками.

Напоминатель – представление сведений об условиях, при которых пользователь должен получать уведомления о предстоящих работах. Помимо воспроизведения традиционно заданных самим пользователем напоминаний, играет важную роль в генерации уведомлений на основе автоматизированных алгоритмов построения плана действий.

Повторитель – содержит сведения о частоте повторения работ. В представленной онтологии является самостоятельной сущностью, что связано с необходимостью реализации более гибких возможностей задания повторений с переменным интервалом.

Классы онтологии. Класс *Пользователь* – служит для представления субъекта, использующего программу-планировщик, базирующуюся на рассматриваемой системе знаний, и обладает двумя слотами и тремя отношениями. Структура класса представлена на рисунке 2, а.

Слоты класса *Пользователь*:

- *имя* – обязательный слот, содержащий уникальное имя субъекта или идентификационные данные;
- *текущая работа* – работа, выполняемая

пользователем в текущий момент.

Отношения класса *Пользователь*:

- *имеет условия* – содержит ссылку на объект условия местонахождения субъекта. С помощью данных сведений можно отслеживать положение пользователя относительно мест, связанных с заданными им работами;

- *задает работы* – ссылка на созданные пользователем объекты-работы;

- *имеет историю* – содержит указатель на индивидуальную историю завершённых работ.

Класс *Список работ* – абстрактный класс, являющийся базовым классом для представления различных коллекций работ – классов *План* и *История*. Обладает одним отношением. Структура класса представлена на рисунке 2, б.

Отношение класса *Список работ*:

- *имеет работы* – ссылка на работы пользователя.

Класс *План* – предназначен для хранения ссылок на предстоящие работы пользователя и обладает одним отношением. Структура класса представлена на рисунке 2, в.

Отношение класса *План*:

- *имеет работы* – ссылка на предстоящие работы определенного пользователя.

Класс *История* – предназначен для хранения ссылок на завершённые работы пользователя с целью повторного использования данных для формирования напоминаний к работам вида III. Обладает одним отношением. Структура класса представлена на рисунке 2, г.

Отношение класса *История*:

- *имеет работы* – ссылка на завершённые работы конкретного пользователя.

Класс *Напоминатель* – при выполнении заданных условий работы формирует для пользователя соответствующее уведомление. Обладает двумя слотами и одним отношением. Структура класса представлена на рисунке 2, д.

Слоты класса *Напоминатель*:

- *автонапоминание* – условие срабатывания напоминателя на основе данных о категории или местоположении работы;

- *напоминание вручную* – конкретные условия, предварительно заданные пользователем.

Отношение класса *Напоминатель*:

- *напоминает пользователю* – ссылка на объект-пользователь, ассоциированный с реальным пользователем, на чей адрес будут отправляться уведомления.

Класс *Работа* – служит для представления работ, создаваемых пользователем. Обладает тремя слотами и четырьмя отношениями. Структура класса представлена на рисунке 2, е.

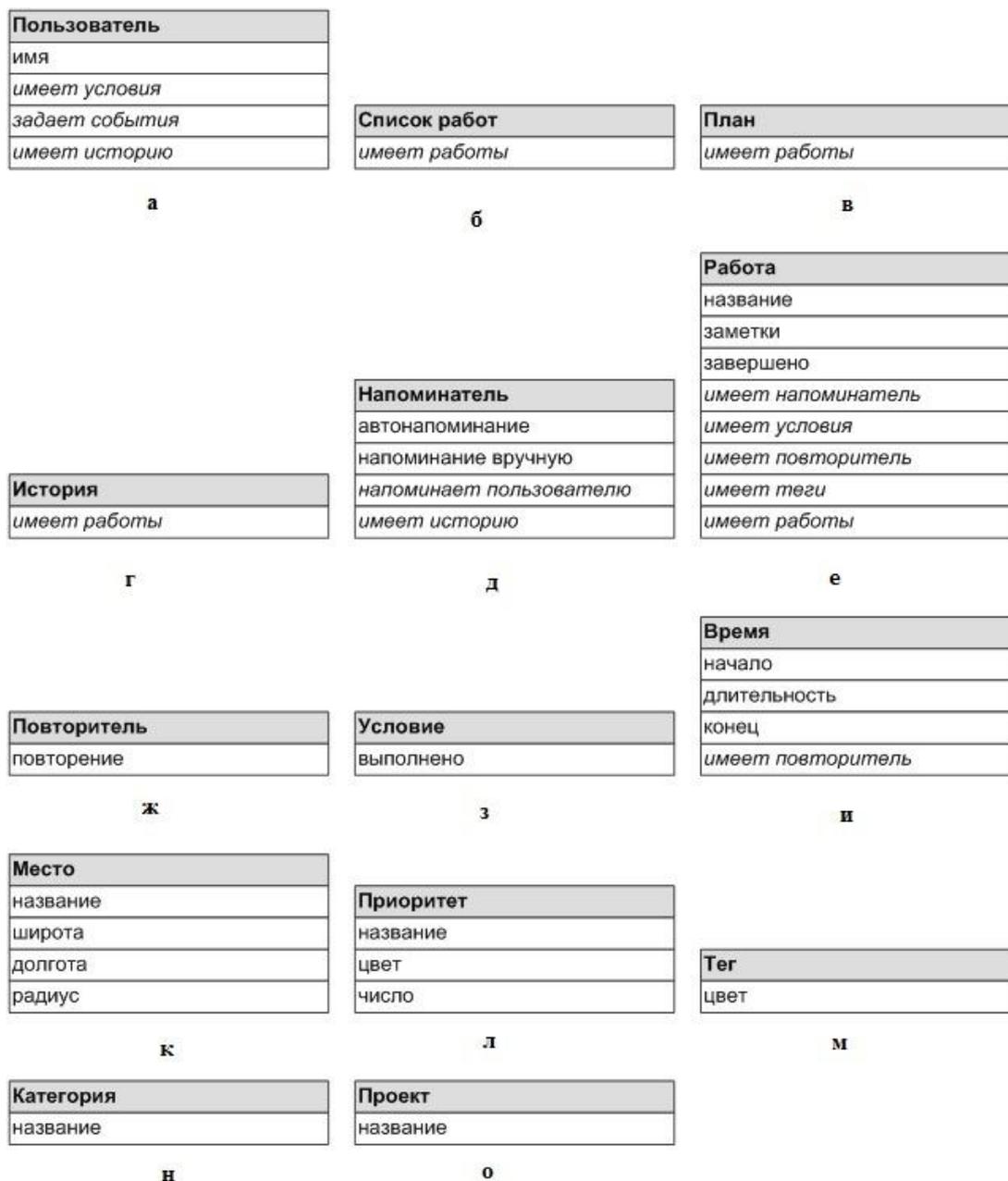


Рисунок 2 – Структуры классов: а – Пользователь; б – Список работ; в – План; г – История; д – Напоминатель; е – Работа; ж – Повторитель; з – Условие; и – Время; к – Место; л – Приоритет; м – Тег; н – Категория; о – Проект

Слоты класса *Работа*:

- *название* – содержит короткое название работы;
- *заметки* – более подробный комментарий о работе;
- *завершено* – определяет факт завершенности работы.

Отношения класса *Работа*:

- *имеет условия* – ссылка на объекты условия работ: время, место и приоритет;
- *имеет напоминатель* – ссылка на объект-напоминатель;
- *имеет теги* – ссылка на объект-катеорию работы;

- *имеет работы* – ссылка на дочерние под-работы.

Класс *Повторитель* – хранит периодичности, с которыми повторяются как отдельные работы, так и части работ вида II внутри их самих. Обладает одним слотом. Структура класса представлена на рисунке 2, ж.

Слот класса *Повторитель*:

- *повторение* – значение периода повторения, заданное пользователем.

Класс *Условие* – абстрактный класс, служащий базовым классом для представления различных видов условий работы – классов *Время*, *Место* и *Приоритет*. Обладает одним слотом.

Структура класса представлена на рисунке 2, з.

Слот класса *Условие*:

- *выполнено* – признак выполнения условия.

Класс *Время* – содержит временные условия работ. Обладает тремя слотами и одним отношением. Структура класса представлена на рисунке 2, и.

Слоты класса *Время*:

- *начало* – момент, в который работа начинается;
- *длительность* – количество времени, которое уделяется работе;
- *конец* – момент, в который работа заканчивается,

Отношение класса *Время*:

- *имеет повторитель* – ссылка на объект-повторитель, который определяет периодичность частей работ вида II внутри себя.

Класс *Место* – хранит условия местоположения работ. Обладает четырьмя слотами. Структура класса представлена на рисунке 2, к.

Слоты класса *Место*:

- *название* – может содержать название места;
- *широта* – в паре со следующим слотом задает положение места работы в пространстве;
- *долгота*;
- *радиус* – используется для автоматического запуска напоминателя при приближении пользователя на заданное расстояние к месту, определяемому значениями слотов *широта* и *долгота*.

Класс *Приоритет* – хранения условия важности работ относительно друг друга. Обладает тремя слотами. Структура класса представлена на рисунке 2, л.

Слоты класса *Приоритет*:

- *название* – значение приоритета для системы приоритетов, основанной на лингвистических оценках, например «важно» или «в первую очередь»;
- *цвет* – для визуального оформления;
- *число* – значение приоритета для числовой системы приоритетов.

Класс *Тег* – абстрактный класс, служащий базовым классом для создания теговых классов, представляющих различные группы работ – *Категория* и *Проект*. Обладает одним слотом. Структура класса представлена на рисунке 2, м.

Слот класса *Тег*:

- *цвет* – используется для визуального оформления.

Класс *Категория* – конкретный тег, представляющий различные группы, к контексту которых относится данная работа. При этом одна и та же работа может быть отнесена к нескольким категориям. Класс обладает одним слотом. Структура класса представлена на рисунке 2, н.

Слот класса *Категория*:

- *название* – название группы работ, например «работа» или «отдых».

Класс *Проект* – конкретный тег, аналогичный по семантике тегу *Категория*, используемый для указания группы работ, предполагающих выполнение некоторых регламентированных (с известным алгоритмом реализации) процессов – технологических или организационных. Одна и та же работа может быть отнесена к нескольким проектам. Класс обладает одним слотом. Структура класса представлена на рисунке 2, о.

Слот класса *Проект*:

- *название* – название группы работ, предполагающих реализацию регламентированного процесса, например «подготовить отчет» или «построить дом».

Алгоритмы планирования. Второй важной составляющей интеллектуальной программы планировщика являются алгоритмы автоматизированного построения плана действий.

При создании алгоритмов принято определять критерий качества и цель реализации этих алгоритмов. Указать вычисляемый или хотя бы оцениваемый критерий качества функционирования планировщика нелегко, а может быть, и невозможно вовсе. Концептуально главным из критериев с интуитивно уясняемой семантикой является уверенность (правдоподобность, вероятность, надёжность) в выполнении сформированного плана. Целью функционирования планировщика является максимизация указанной уверенности. Конечно же, можно вести речь и о многочисленных альтернативах, таких как длительность выполнения плановых заданий, длительность бездействия исполнителя плана и тому подобное.

Казалось бы, в качестве измеряемого критерия качества планировщика можно использовать количество просроченных плановых заданий, а в качестве цели – минимизацию этого количества. Но важно понимать, что указанный критерий зависит не только от особенностей планировщика, но и от активности (добросовестности, мотивированности, нацеленности) субъекта, использующего этот планировщик.

В процессе разработки алгоритмов была создана система классификации работ относительно их временных характеристик и выделены три вида работ:

- I) имеющие только момент начала;
- II) имеющие моменты начала и завершения;
- III) не имеющие временных границ.

Первый вид работ уже реализован в современных планировщиках. Так как у этих дел определен момент начала, то это наиболее четко оп-

ределенные во времени работы пользователя, которые будут служить основанием для формирования динамического плана действий путем включения в него работ второго и третьего классов, которые могут быть сдвинуты относительно своих первоначальных временных границ. Работы вида I неразрывны и точно определены во времени. Таким образом, они изначально включены в план действий пользователя и планировщику не требуется самому выбирать время напоминания, следить за разбивкой работы на части и устранять конфликтные ситуации.

Только пользователь напрямую может изменять временные рамки и прочие атрибуты работ этого вида. Единственным главным значением здесь является текущее время, которое планировщик сравнивает с атрибутами всех работ вида I. Подобные алгоритмы повсеместно применяются во всех современных планировщиках [3], позволяя успешно обрабатывать работы данного вида. Как следует из рассуждений выше, никакой интеллектуальной составляющей для реализации таких планировщиков не требуется. Планировщик в данном случае четко следует заданным пользователем инструкциям. Процедуры устранения конфликтов работ и корректировки плана будут описаны позже.

Условная графическая модель выполнения некоторого плана, состоящего из работ вида I, представлена на рисунке 3, а. Стрелки на этом рисунке символизируют моменты начала выполнения работ, а последующие примыкающие прямоугольники – длительности выполнения работ. Как видно из рисунка, между работами могут оставаться достаточно большие участки свободного времени, в которые пользователь должен самостоятельно принимать решения, планируя использование свободного времени вручную.

Работы вида II обладают следующими особенностями: 1) их длительность может быть существенно меньше запланированного под эти работы промежутка времени; 2) они могут выполняться с перерывами; 3) они должны быть выполнены в рамках запланированных временных границ. Например, написание некоторого документа может продолжаться целый день с перерывами на отдых, прием пищи, выполнение иных срочных дел. Поэтому эта категория задач обладает двойными характеристиками. С одной стороны, пользователь определяет временные границы, а с другой – контроль выполнения работы по частям осуществляет планировщик. Фактически работы любых видов могут в итоге превысить отведенное на них время, но при планировании заранее такие ситуации абсурдны.

Стратегия планирования работ вида II основывается на анализе оставшегося до завершения задачи времени. Вначале алгоритм посылает напоминание пользователю о начале работы подобно ситуации с делами I вида.

Как только пользователь или непосредственно планировщик прерывает (но не завершает) выполнение задания (например, для отдыха или осуществления работ другого вида), алгоритм через определенные интервалы времени начинает предлагать возобновить прерванную работу. Пользователь может многократно возвращаться и вновь прерывать выполнение. Как только время, необходимое на выполнение оставшейся части работы, сравнивается с разностью момента завершения и текущего момента времени, планировщик сообщит об этом пользователю и далее будет поставлен вопрос уже о переносе срока завершения задачи. Так как работы вида II позволяют разбивать время своего выполнения на части, то при составлении плана допускают перекрытие с работами вида I. Контроль времени в таких случаях также осуществляет планировщик.

Возможный вариант работы (протокол) алгоритма в виде графической схемы приведен на рисунке 3, б. Видно, что использование работ вида II позволяет лучше организовать время пользователя, равномерно распределяя как работы небольшой длительности, так и длительные монотонные дела.

Работы вида III в настоящее время являются наиболее проблемной категорией, поскольку в современных планировщиках средств для их обработки нет.

Алгоритм планировщика должен самостоятельно принимать решение о внесении той или иной работы вида III в план действий пользователя, исходя из контекста ситуации:

свободное время в плане – время между окончанием предыдущей работы и началом следующей работы вида I или II. Если для работы указано местоположение, то время, необходимое для достижения заданной точки, не должно превышать свободный отрезок плюс время, необходимое для прибытия в точку следующей работы вида I или II;

- местоположение (s) – если для работы указано местоположение, то при прочих равных условиях в план следует включать работу с ближайшим местоположением;

- категория (c) – на основе анализа истории завершённых работ планировщик должен сделать вывод о предпочтениях пользователя, то есть работы какой категории пользователь чаще делает в текущий временной отрезок. Естествен-

но, предпочтение на включение в план следует отдать работе с наиболее похожими категориями;

- приоритет (p) – несмотря на отсутствие временных рамок, пользователь вполне конкретно может определить категорию важности той или иной работы. Предпочтение следует отдавать работам с наиболее высоким приоритетом.

Для расчета количественного показателя наиболее подходящей по ситуации работы предлагается формула, в которой каждый контекстный показатель (кроме времени) нормирован и умножен на коэффициент его важности, определяемый пользователем:

$$I_i = k_s \frac{s_i}{s_{max}} + k_c \frac{c_i}{c_{max}} + k_p \frac{p_i}{p_{max}},$$

где: I_i – коэффициент применимости i -й работы в текущей ситуации;

s_i – расстояние от места i -й работы до пользователя;

s_{max} – наибольшее s_i среди всех участвующих в расчетах работ;

c_i – число совпадений категорий i -й работы с категориями уже выполненными в это же время работ;

c_{max} – наибольшее c_i среди всех участвующих в расчетах работ;

p_i – значение приоритета i -й работы;

p_{max} – наибольшее p_i среди всех участвующих в расчетах работ;

k_s, k_c, k_p – коэффициенты важности параметров s, c и p соответственно.

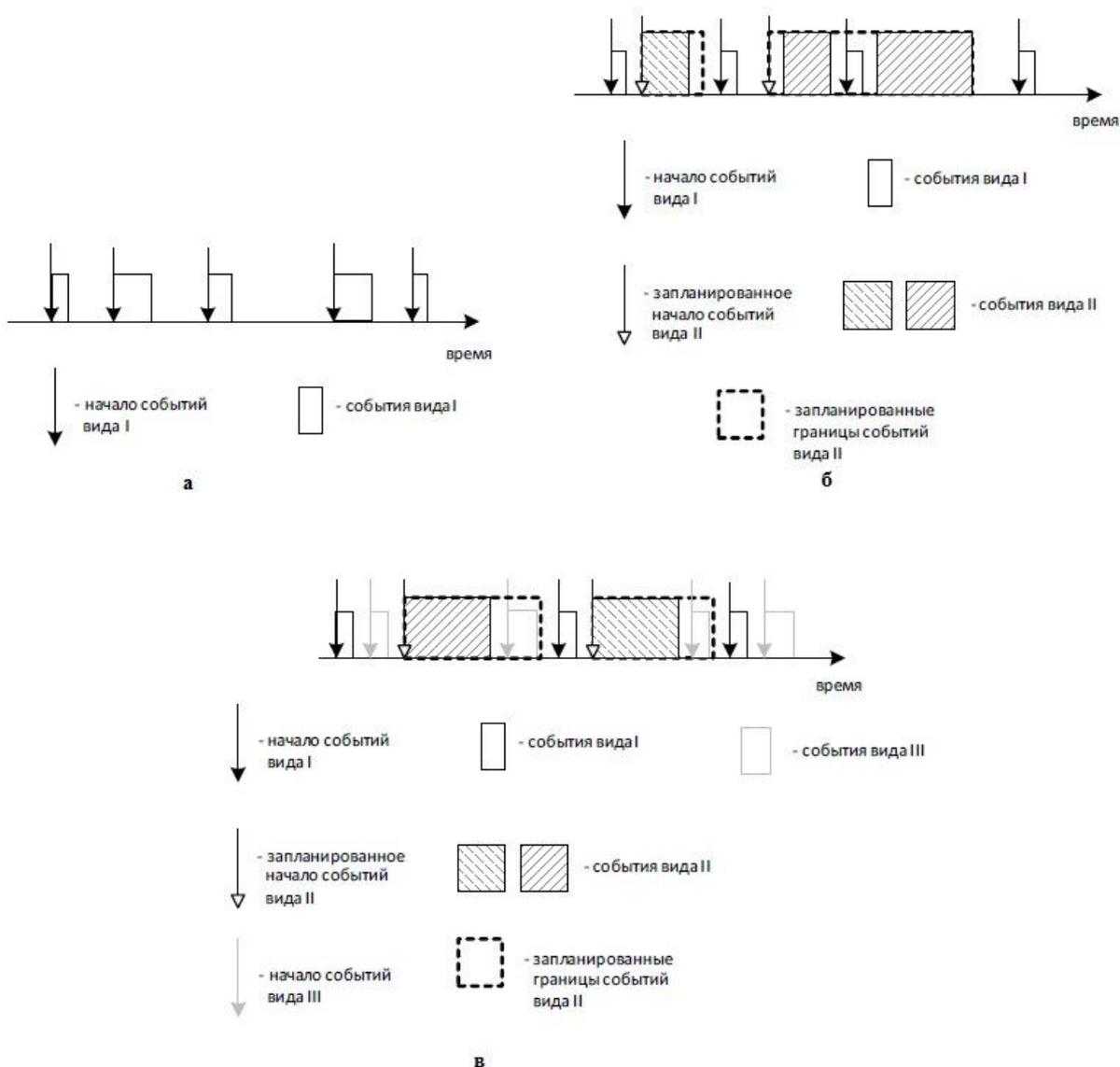


Рисунок 3 – Результаты выполнения плана работ вида: а – I; б - I и II; в - I, II и III

Алгоритм выборки наиболее подходящей работы для включения в план можно описать следующим образом. На первом этапе рассчиты-

вается свободное время в плане и отбираются работы вида III, проходящие по временному критерию, как это описано выше. Если у работы

не указано местоположение, то оно включается в отбор. Далее вычисляется индекс соответствия I для каждой отобранной работы. Причем если местоположение работы не указано, то нормированное значение приравнивается к единице, если не указаны категория или приоритет, то – к нулю. Далее работы сортируются по убыванию индекса и предлагаются пользователю в качестве текущей работы.

Возможный вариант выполнения плана, включающего работы всех трёх видов, показан на рисунке 3, в. Работы вида III способны заполнить практически все свободные промежутки времени в плане пользователя. Таким образом, при включении в план работ такого рода мы получаем максимально возможное количество рабочего времени, потраченного на выполнение заданий. Это, несомненно, положительно скажется на производительности труда пользователя, позволит увеличить количество выполненных дел и минимизировать время бездействия, а также избавит от необходимости тщательного построения ежедневного плана действий и его динамической корректировки во время выполнения.

Очевидно, что реализация поддержки работ вида III является важным улучшением и необходима любому современному планировщику.

Во время формирования плана действий могут возникнуть следующие конфликтные ситуации:

- 1) работы вида I накладываются друг на друга;
- 2) работы вида II накладываются друг на друга;
- 3) работы вида III накладываются друг на друга;
- 4) работы вида I накладываются на работы вида II (и наоборот);
- 5) работы вида I накладываются на работы вида III (и наоборот);
- 6) работы вида II накладываются на работы вида III (и наоборот).

Рассмотрим способы устранения вышеописанных конфликтных ситуаций.

Работы вида I накладываются друг на друга. Предпочтение отдается работе с высшим приоритетом. Если приоритеты одинаковы/ отсутствуют, то решает пользователь.

Работы вида II накладываются друг на друга. Пока одну из работ можно отложить в пользу другой, планировщик делает это без вмешательства пользователя, иначе конфликт устраняется, как в случае I.

Работы вида III накладываются друг на друга. Такая ситуация возможна при задании авто-

напоминаний. Конфликт устраняется, как в случае I.

Работы вида I накладываются на работы вида II (и наоборот). Пока работу вида II можно отложить в пользу работы вида I, планировщик делает это без вмешательства пользователя, иначе конфликт устраняется, как в случае I.

Работы вида I накладываются на работы вида III (и наоборот). Предпочтение отдается работе вида I.

Работы вида II накладываются на работы вида III (и наоборот). Пока работу вида II можно отложить в пользу работы вида III, планировщик делает это без вмешательства пользователя, иначе предпочтение отдается работе вида II.

Перед тем как добавить работу вида I или II в список планируемых пользователем действий (или поменять его временные характеристики), планировщику следует проверить доступность заданных временных рамок. При возникновении конфликта он решается в соответствии правилами, описанными выше.

Поскольку добавление работ вида III не предполагает их непосредственного размещения во времени, то никаких дополнительных действий по обработке не требуется. Работа просто помещается в список планируемых работ.

Удаление невыполненных работ всех видов не является предметом совершенствования и представляет собой освобождение временных рамок, отведенных под работы пользователем.

Завершение выполненных работ предполагает их последующее размещение в отдельной коллекции класса *История*. Впоследствии на основе статистики о выполненных работах осуществляется ранжирование работ вида III по степени соответствия текущей ситуации.

Заключение. Предложенные структуры данных позволяют осуществлять представление работ всех разновидностей относительно временных характеристик в унифицированном виде, а также хранить всю необходимую сопутствующую информацию для работы с ними. Разработанные алгоритмы призваны обеспечить интеллектуальную составляющую автоматизированных систем индивидуального планирования. Это достигается путем включения в план действий работ с неопределенными временными характеристиками согласно описанным методам.

Для оценки потенциального прироста эффективности деятельности пользователей, ожидаемого от использования программы планировщика с реализацией предложенных методов, был проведен опрос. В нем приняли участие 30 респондентов из различных сфер организации времени: работающие, а также люди с норми-

рованным и ненормированным рабочим днем. Респондентам предлагалось ответить на вопрос: сколько времени в день они могли бы потратить на отложенные «на потом» дела, если бы им вовремя об этом подсказывали. Это и будет время, которое может быть потрачено продуктивно при использовании программ планировщиков с реализацией предложенных методов. В качестве общей продолжительности дня было принято значение – 16 часов. Результаты приведены в таблице.

Результаты опроса

Кол-во респондентов	Время, часы
17	2
9	3
4	4

Как видно из результатов опроса, средний показатель по всем респондентам: 2.57 часа. Из этого можно вычислить средний процентный прирост эффективности $\Delta \varepsilon_{cp}$:

$$\Delta \varepsilon_{cp} = \frac{2,57 * 100}{16} \approx 16 \% .$$

Также, если делать расчет по минимальному $\Delta \varepsilon_{min}$ и максимальному $\Delta \varepsilon_{max}$ значениям опроса, получим предельные значения прироста эффективности:

$$\Delta \varepsilon_{min} = \frac{2 * 100}{16} \approx 12,5 \% ,$$

$$\Delta \varepsilon_{max} = \frac{4 * 100}{16} \approx 25 \% .$$

Таким образом, потенциальный прирост эффективности планирования при использовании программ планировщиков с реализацией пред-

ложенных методов может быть на 12,5-25 % больше по сравнению с существующими аналогами.

Данный способ дает оценку потенциального прироста эффективности деятельности субъекта при использовании планировщика, поддерживающего работы вида III, – с неопределёнными временными характеристиками. Реальное повышение эффективности деятельности субъекта будет существенно зависеть от дисциплинированности и состояния этого субъекта, так как в конечном счёте именно он выполняет или меняет предложенный планировщиком план действий. В качестве дальнейшего развития предложенных методов можно предложить использование биометрических параметров пользователя в интеллектуальных алгоритмах построения плана действий.

Библиографический список

1. *Архангельский Г. А.* Тайм-драйв: как успевать жить и работать. 8-е изд. М.: Манн, Иванов и Фербер, 2008. 256 с.
2. *Allen D.* Getting Things Done: The Art of Stress-Free Productivity // Penguin Group (USA) Incorporated. 2001. 267 pages.
3. *Белов В.В., Наумович М.В.* Проблемы реализации функциональности событий в программных средствах индивидуального планирования // Вестник РГРТУ. Рязань, 2011. Вып. 36. С. 43 – 51.
4. *Джарратано Дж., Райли Г.* Экспертные системы: принципы разработки и программирование // М.: Вильямс, 2006. 1152 с.
5. *Кандрашина Е.Ю., Литвинцева Л.В., Поспелов Д.А.* Представление знаний о пространстве и времени в системах искусственного интеллекта. М.: Наука, 1988. 328 с.