

УДК 519.688

Древовидная модель решения многомерных мультимодальных оптимизационных задач гибридными методами поиска

П. А. Чернов*

В статье рассматривается проблема сочетания методов поиска на примере генетических алгоритмов и детерминированных методов локального поиска с целью использования сильных сторон обоих подходов для решения многомерных мультимодальных оптимизационных задач. Для формализации правил сочетания генетических алгоритмов и методов локального поиска предложена древовидная модель решения оптимизационной задачи, определен алгоритм обхода дерева. На основе предложенной древовидной модели, генетических алгоритмов и детерминированных методов локального поиска предложен гибридный метод решения многомерных мультимодальных оптимизационных задач.

Ключевые слова: древовидная модель решения оптимизационной задачи, генетический алгоритм, метод локального поиска, метод многомерной оптимизации, гибридный метод поиска.

Введение

Среди методов решения оптимизационных задач (ОЗ) выделим методы детерминированного локального поиска (ЛП) [1, 2] и генетические алгоритмы (ГА) [3 — 5]. Среди практически важных свойств ГА отметим возможность одновременной работы с множеством решений ОЗ. Поскольку оценка разных особей популяции может быть выполнена независимо, то возможно параллельное выполнение такой оценки. Данный факт особенно важен, если учесть, что современные компьютеры, как правило, оснащены многоядерными процессорами. Другое полезное свойство ГА обусловлено простотой их реализации в случае многомерных ОЗ — увеличение размерности ОЗ часто приводит лишь

к удлинению кодирующей решение ОЗ хромосомы. Более того, если рост размерности ОЗ часто приводит к резкому снижению эффективности детерминированных методов поиска [1], то ГА подвержены этому в меньшей степени. Еще одним важным свойством ГА является их применимость для решения мультимодальных ОЗ — с помощью ГА можно получить решения из множества локальных оптимумов, причем ГА имеют средства выхода из локальных оптимумов, чтобы не допустить преждевременной сходимости. Решая многомерную мультимодальную задачу, ГА может сравнительно быстро найти хорошие решения во множестве окрестностей локальных оптимумов, однако нахождение собственно локальных оптимумов оказывается для него затруднительным. В свою очередь методы ЛП такие, как метод дихотомии, золотого сечения [2] и другие, способны сравнительно быстро находить локальный оптимум, но не приспособлены для поиска глобального оптимума.

* pavel_chernov_articles@mail.ru

ма. Учитывая сказанное выше, становится актуальным вопрос сочетания ГА и методов ЛП, использования сильных сторон обоих в гибридных методах поиска [5].

В данной статье предложена древовидная модель решения многомерных мультимодальных ОЗ, которая может быть использована для формализации правил сочетания ГА и методов ЛП.

Постановка оптимизационной задачи

Прежде чем рассмотреть предлагаемую древовидную модель решения оптимизационной задачи (ДМРОЗ), определим, как ОЗ должна быть поставлена. Это связано с тем, что одним из наиболее существенных факторов, обуславливающих успешность применения ГА, является способ кодирования параметров оптимизационной задачи (ПОЗ). В частности, желательно, чтобы кодирование допускало как можно меньше недопустимых решений.

Пусть ПОЗ *poz* может принимать одно из *nVals* значений. Пусть каждому значению ПОЗ соответствует фенотип и генотип. Фенотип значения ПОЗ является исходным представлением значения ПОЗ, генотип — представлением значения ПОЗ, удобным для ГА. Тогда каждому *poz* можно сопоставить нумерованное множество фенотипов значений ПОЗ $vpt = \{vpt_1, \dots, vpt_{iVal}, \dots, vpt_{nVals}\}$, нумерованное множество соответствующих генотипов значений ПОЗ $vgt = \{vgt_1, \dots, vgt_{iVal}, \dots, vgt_{nVals}\}$, множество номеров значений ПОЗ $vnum = \{vnum_1, \dots, vnum_{iVal}, \dots, vnum_{nVals}\}$, где *iVal* — номер значения ПОЗ, $1 \leq iVal \leq nVals$. Генотип vgt_{iVal} значения ПОЗ представляет собой двоичный код номера *iVal* значения ПОЗ и может состоять из $nGenes = \lceil \log_2(nVals) \rceil$ генов, каждый ген принимает значения из множества {0, 1}. Может быть полезным потребовать, чтобы значения во множестве *vpt* были пронумерованы по возрастанию или по убыванию этих значений. На рис. 1 проиллюстрированы множества фенотипов *vpt*, генотипов *vgt*, номеров *vnum* значений ПОЗ и отношения между ними.

Пусть *nPoz* — число варьируемых ПОЗ, $nPoz \geq 1$. Определим нумерованное множество ПОЗ $poz = \{poz_1, \dots, poz_{iPoz}, \dots, poz_{nPoz}\}$, где *iPoz* — порядковый номер ПОЗ в *poz*. Множество номеров ПОЗ обозначим $poznum = \{poznum_1, \dots, poznum_{iPoz}, \dots, poznum_{nPoz}\}$. Порядок ПОЗ в *poz*, определяемый нумерацией *poznum*, важен, поскольку такой порядок определяет взаимное расположение генов в хромосоме, имеющее большое значение потому, что ГА оперируют шаблонами [3] (схемами [4], шимами [5]).

Решение ОЗ представимо особью. Пусть генотип особи образован одной хромосомой. В свою очередь хромосома образована генотипами ПОЗ в порядке расположения этих ПОЗ в *poz*. Аналогично, фенотип особи образован фенотипами ПОЗ из *poz*.

Запишем решение ОЗ в виде нумерованного множества $sol = \{vpt_1, \dots, vpt_{iPoz}, \dots, vpt_{nPoz}\}$, где vpt_{iPoz} — фенотип некоторого значения из poz_{iPoz} . Пространство поиска *S* можно определить как декартово произведение множеств фенотипов значений ПОЗ — $S = \times_{iPoz=1}^{nPoz} poz_{iPoz} \cdot vpt$.

Пусть определена модель объекта оптимизации, которая позволяет проверить, является ли решение *sol* допустимым в условиях некоторых заданных ограничений *R*. Обсуждение модели объекта оптимизации выходит за рамки данной статьи. Пусть также определена целевая функция (ЦФ) *tf*, такая что $tf(sol) > 0$. Тогда модель ОЗ можно записать в виде $MOZ = \langle S, R, tf \rangle$. Требование оптимизации $tf(sol) \rightarrow \max$, то есть требуется найти $sol^* \in S$, удовлетворяющее *R*, такое, что $f(sol^*) = \max_{sol \in S} tf(sol)$.

Процедуры ГА- и ЛП-оптимизации

Согласно [5] сочетать ГА и методы ЛП можно следующим образом. Пусть с помощью ГА получена некоторая текущая популяция. К каждой особи текущей популяции можно применить метод ЛП, по результатам работы которого для каждой особи может быть сформирована новая особь, соответствующая локаль-



Рис. 1. Множества ПОЗ, номеров ПОЗ, фенотипов, генотипов, номеров значений ПОЗ и отношения между ними

ному оптимуму, в окрестности которого находилась исходная особь до применения метода ЛП. Так формируется новая популяция особей. Поскольку ГА порождает множество особей, располагающихся в окрестностях множества локальных оптимумов, то после применения метода ЛП высока вероятность попадания особей популяции в глобальный оптимум.

Процесс создания новой популяции методом ЛП будем называть процедурой ЛП-оптимизации, а с помощью ГА — процедурой ГА-оптимизации. На рис. 2 приведены примеры схем процедур ГА- и ЛП-оптимизации. Кратко опишем данные схемы.

Пусть процедура ГА-оптимизации реализована схожим с классическим ГА [4] образом и может быть представлена следующими шагами:

1. Формирование родительской популяции, то есть среди особей текущей популяции выполняем отбор особей в родительскую популяцию, которая будет формировать популяцию потомков.
2. Скрещивание особей выполняется попарно, формируется по две новые особи на родительскую пару.

Новые особи образуют так называемую популяцию потомков

3. Мутация. Особи популяции потомков претерпевают инвертирование значений генов, выбираемых случайным образом.

4. После мутации популяция потомков полностью переходит в состав новой текущей популяции.

5. Для каждой особи текущей популяции рассчитываем ЦФ.

Работа изложенной выше процедуры ГА-оптимизации достаточно очевидна, когда в процессе оптимизации участвуют все ПОЗ из poz . Предположим, что процедуру ГА-оптимизации необходимо выполнить для множества $pozA$ активных ПОЗ, $pozA \subseteq poz$. Обозначим $pozB = poz \setminus pozA$ множество неактивных ПОЗ. Фенотипы, гены, участки хромосом, соответствующие $pozB$, также будем называть неактивными, соответствующие $pozA$ — активными (рис. 3, неактивные ПОЗ закрашены серым цветом).

Те ПОЗ, которые являются на текущем этапе оптимизации неактивными, на предшествующих этапах оптимизации могли быть активными. Тогда одним из

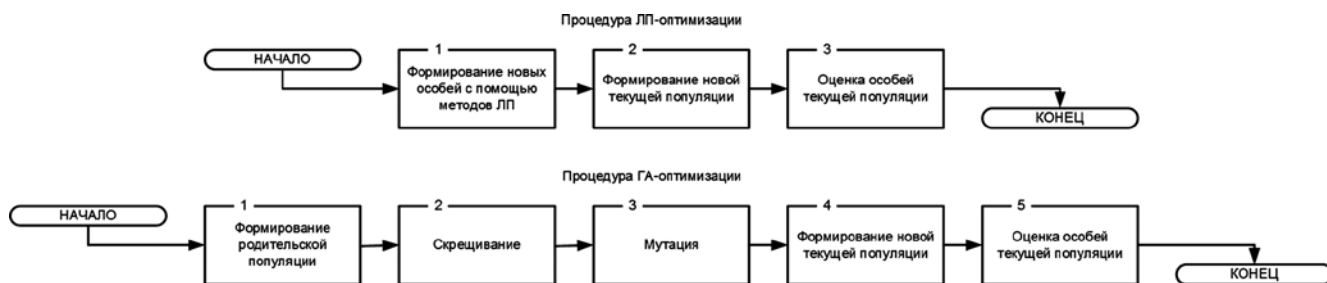


Рис. 2. Процедуры ГА- и ЛП-оптимизации

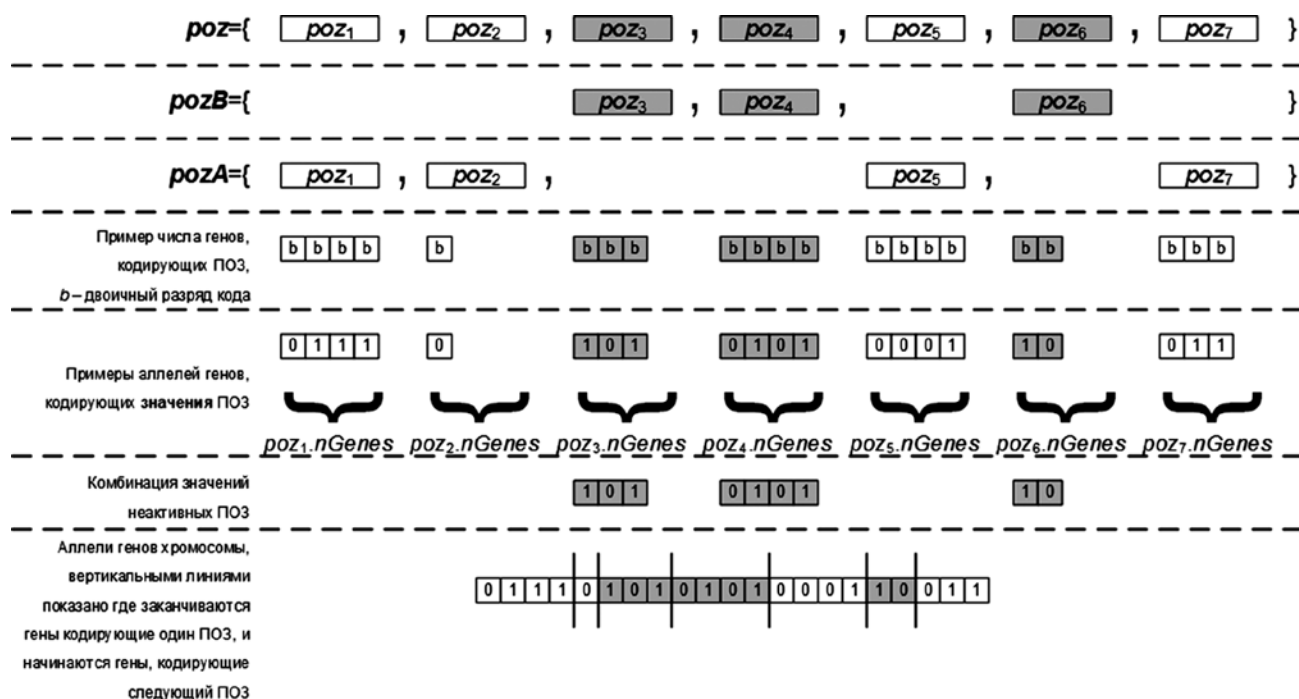


Рис. 3. Примеры множеств активных и неактивных ПОЗ

результатов работы предшествующих этапов оптимизации можно считать набор комбинаций значений неактивных на текущем этапе оптимизации ПОЗ (рис. 4). Можно потребовать, чтобы этот набор комбинаций был представлен в популяции, которая будет сформирована по завершении очередного этапа оптимизации. Приведем некоторые соображения о том, какие изменения может потребоваться внести в работу процедуры ГА-оптимизации.

Во-первых, можно оставить структуру хромосомы неизменной (включающей как активные участки, так и неактивные) и модифицировать только генетические операторы (ГО) процедуры ГА-оптимизации, например, согласно следующим соображениям.

— В результате применения ГО селекции на базе, скажем, метода рулетки для отбора особей из исходной популяции в родительскую набор комбинаций значений неактивных ПОЗ (КЗНПОЗ) в родительской популяции может отличаться от набора КЗНПОЗ исходной популяции. Исправить это можно, например, назначив особям родительской популяции КЗНПОЗ из набора исходной популяции. Детальное обсуждение алгоритма назначения выходит за рамки данной статьи, отметим лишь, что такое назначение можно выполнить, например, случайным образом.

— Для ГО скрещивания можно потребовать, чтобы он не выполнял разрыв в неактивных участках хромосомы. Однако если КЗНПОЗ определена на $pozB$ и $\exists poz_j \in pozB, \exists poz_i \in pozA, \exists poz_k \in pozB \mid j < i < k$, где i, j, k — номера ПОЗ в poz , то набор КЗНПОЗ в популяции после выполнения ГО скрещивания может измениться, и потребуются его восстановить, например, как это было сделано в случае с ГО селекции. Избежать необходимости восстановления набора КЗНПОЗ можно, если потребовать, чтобы разрыв не выполнялся между первым и последним неактивными генами в хромосоме

(для случая, когда, например, две родительские особи образуют двоих потомков по аналогии с ГО скрещивания в классическом ГА).

— Для ГО мутации можно потребовать, чтобы он не изменял гены в неактивных участках хромосомы.

Таким образом, преобразования в процедуре ГА-оптимизации могут приводить к изменениям только лишь генов, соответствующих активным ПОЗ, но для вычисления ЦФ необходимы значения всех ПОЗ, в том числе неактивных.

Недостаток случайного назначения КЗНПОЗ особям результирующей (родительской, потомков и т.п.) популяции в том, что сравнительно высока вероятность разрушения шаблонов. Чтобы отчасти исправить это, например, можно было бы некоторым образом учитывать схожесть особей исходной и результирующей популяций. Отметим, что потеря шаблонов свойственна и ГО в классическом ГА.

Во-вторых, можно изменить структуру хромосомы, оставив только активные участки. Тогда можно не модифицировать ГО селекции, скрещивания и мутации в процедуре ГА-оптимизации, и после ГО мутации выполнить назначение особям популяции потомков КЗНПОЗ из соответствующего набора КЗНПОЗ исходной популяции, например, случайным образом.

Другой способ выполнения процедуры ГА-оптимизации при хромосоме, содержащей только активные участки, состоит в том, чтобы для каждой особи текущей популяции (и соответствующей ей КЗНПОЗ) создать новую популяцию такую, что генотип особей новой популяции будет образован только генотипами ПОЗ из $pozA$. В таком случае суммарное число эволюционирующих особей увеличивается в разы, что может быть проблематичным, например, в условиях, когда расчет ЦФ является вычислительно сложным, и поэтому далее в статье не рассматривается.

Пусть процедуру ЛП-оптимизации можно представить следующими шагами.

Пример популяции

Номер ПОЗ	Генотип особи																					Генотипы набора КЗНПОЗ популяции									
	1				2				3				4				5				6				3		4		6		
Лocus	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	6	7	8	9	10	11	12	17	18	
Номер особи в популяции	1	0	1	1	1	0	0	1	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0
	2	0	0	1	1	1	1	0	1	1	1	0	1	0	0	1	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0
	3	1	1	1	1	0	1	1	1	0	1	1	1	1	0	0	1	0	0	0	1	1	1	1	1	0	1	1	1	0	0
	4	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1	0

Рис. 4. Пример генотипов набора КЗНПОЗ популяции

1. Для всех особей активен только один ПОЗ poz_{iPoz} . В результате применения метода ЛП для особи вычисляется новое значение активного варьируемого ПОЗ (пробная точка), которое может быть округлено до ближайшего значения в $poz_{iPoz} \cdot vpt$. При этом может быть полезным сохранить неокругленное значение пробной точки. Неактивные гены хромосомы новой особи формируются из соответствующих генов хромосомы исходной особи, активные гены (соответствующие poz_{iPoz}) определяются генотипом найденного нового значения варьируемого ПОЗ.

2. Полученные с помощью метода ЛП особи образуют новую текущую популяцию.

3. Для каждой особи текущей популяции рассчитываем ЦФ.

Предполагается, что в рамках процедуры ЛП-оптимизации активен только один ПОЗ из poz . Пусть на каждой итерации применения процедуры ЛП-оптимизации вычисляется значение ЦФ только в одной пробной точке. Часто методы ЛП для выполнения очередного шага используют значения ЦФ в нескольких последних пробных точках. Тогда будем сохранять значения ЦФ и пробные точки по мере их вычисления.

Древовидная модель решения оптимизационной задачи

После того, как алгоритмы реализующие процедуры ГА- и ЛП-оптимизации некоторым образом определены, необходимо формализовать правила применения этих процедур. Предлагается использовать ДМРОЗ $D\bar{M}R\bar{O}Z$, в которую входят дерево $T\bar{D}M\bar{R}\bar{O}Z$ и алгоритм обхода $TADMROZ$.

Определим корневое дерево $TDMROZ$ с порядком на вершинах (детях) [6]:

$$TDMROZ = \langle V, E \rangle,$$

где V — множество вершин; E — множество ребер дерева. Пусть множество вершин V образовано множеством VL листовых вершин (не имеющих потомков) и множеством VNL нелистовых вершин (имеющих потомков), $VNL \neq \emptyset$. Корневую вершину (не имеющую родителей) обозначим VR , $VR \in VNL$. Предполагается, что в $TDMROZ$ содержится более одной вершины.

$$VL = \{VL_{it,ivt}\}; VNL = \{VNL_{it,ivt}\}; V = VL \cup VNL,$$

где it — номер уровня, на котором расположена вершина; ivt — номер вершины на уровне it ; VL — листовая вершина; VNL — нелистовая вершина.

Листовая вершина VL содержит:

- режим оптимизации om ;
- нумерованное множество ПОЗ $pozI$;
- ЦФ tf ;
- критерий останова (КО) SC ;
- указатель на родителя PV .

Таким образом:

$$VL = \langle om, pozI, tf, SC, PV \rangle;$$

$$pozI \subseteq \{poz_1, \dots, poz_{iPoz}, \dots, poz_{nPoz}\};$$

$$om \in \{GA, LP\},$$

где GA, LP — режимы оптимизации, использующие ГА и методы ЛП соответственно, число режимов может быть увеличено.

Примем ограничение, что в режиме ЛП-оптимизации может быть активным только один ПОЗ, в режиме ГА-оптимизации могут быть активными несколько ПОЗ.

Для определенности будем считать, например, что SC выполнен, если выполнен хотя бы один sc :

$$SC = \bigvee_{isc=1}^{nsc} sc_{isc},$$

где sc — элементарный КО (ЭКО); isc — номер ЭКО; nsc — число ЭКО.

Перечислим некоторые из возможных ЭКО, которые обычно применяются для ГА (ЭКО для методов ЛП также можно найти в специальной литературе):

- число созданных особей (либо популяций) превысило заданное значение;
- время выполнения поиска превысило заданное значение;
- отсутствие улучшения ЦФ по результатам создания заданного числа особей (либо популяций);
- отсутствие улучшения ЦФ в течение заданного интервала времени.

Например, с помощью ЭКО можно управлять долями времени, которые приходятся на выполнение различных процедур оптимизации. Могут быть введены и другие ЭКО.

Нелистовая вершина VNL содержит:

- КО SC ;
- указатель на родителя PV ;
- список указателей на потомков DVL (список потомков);
- указатель на текущего потомка $curDV$.

Таким образом:

$$VNL = \langle SC, PV, DVL, curDV \rangle.$$

Поясним, почему в ДМРОЗ в вершинах VL введена возможность определения ЦФ. Дело в том, что среди недостатков ГА отмечают низкую эффективность ГА при решении ОЗ с вычислительно сложным расчетом ЦФ [5]. Предположим, что в процессе оптимизации ЦФ tf по некоторым ПОЗ можно заменить ЦФ tf^* с вычислительно сложным расчетом на приближенную ЦФ tf^* с вычислительно менее сложным расчетом. Тогда, модифицировав конкретную ДМРОЗ, можно определить на ограниченном множестве ПОЗ отдельную оптимизационную подзадачу с ЦФ tf^* . Решение такой оптимизационной подзадачи можно принять в качестве опорного решения исходной ОЗ. Одним из свойств ГА является приспособленность для решения изменяю-

щихся во времени ОЗ, другими словами, после нахождения опорного решения достаточно заменить ЦФ tf^* на tf и продолжить поиск.

Определим алгоритм *TADMROZ* обхода дерева. За основу возьмем алгоритм поиска в глубину [7]. Опишем алгоритм обхода ДМРОЗ, схема которого приведена на рис. 5.

1. В качестве текущей вершины $curV$ выбираем корневую вершину VR , не имеющую родителя ($curV.PV = 0$).

2. Проверяем тип текущей вершины $curV$. Если $curV \in VNL$, то переходим к шагу 3, иначе если $curV \in VL$, то переходим к шагу 5.

3. В качестве текущего потомка текущей вершины $curV.curDV$ выбираем первого потомка из списка потомков текущей вершины $curV.DVL$ ($curV.curDV = curV.DVL$). Предполагается, что $curV.DVL$ указывает на первого потомка из списка.

4. В качестве текущей вершины $curV$ принимаем текущего потомка текущей вершины $curV.curDV$ ($curV = curV.curDV$). Переходим к шагу 2.

5. Выводим для текущей вершины: $curV.om$, $curV.pozl$, $curV.SC$. Вводим параметры, необходимые для расчетов КО.

6. Вычисляем КО текущей вершины $curV.SC$.

7. Если КО $curV.SC$ выполнен, то переходим к шагу 8, иначе переходим к шагу 5.

8. Инициализируем КО текущей вершины $curV.SC$, что может потребоваться поскольку КО был выполнен.

9. В качестве текущей вершины $curV$ принимаем родителя текущей вершины $curV.PV$ ($curV = curV.PV$). Родителем по отношению к листовой вершине может быть только нелистовая вершина. Таким образом, листовая вершина передает управление родителю сразу после выполнения КО.

10. Вычисляем КО текущей вершины $curV.SC$. Таким образом, КО вычисляем независимо от того, является ли текущий потомок $curV.curDV$ последним в списке $curV.DVL$ или нет.

11. Если КО $curV.SC$ выполнен, то переходим к шагу 14, иначе переходим к шагу 12.

12. Проверяем, является ли текущий потомок текущей вершины $curV.curDV$

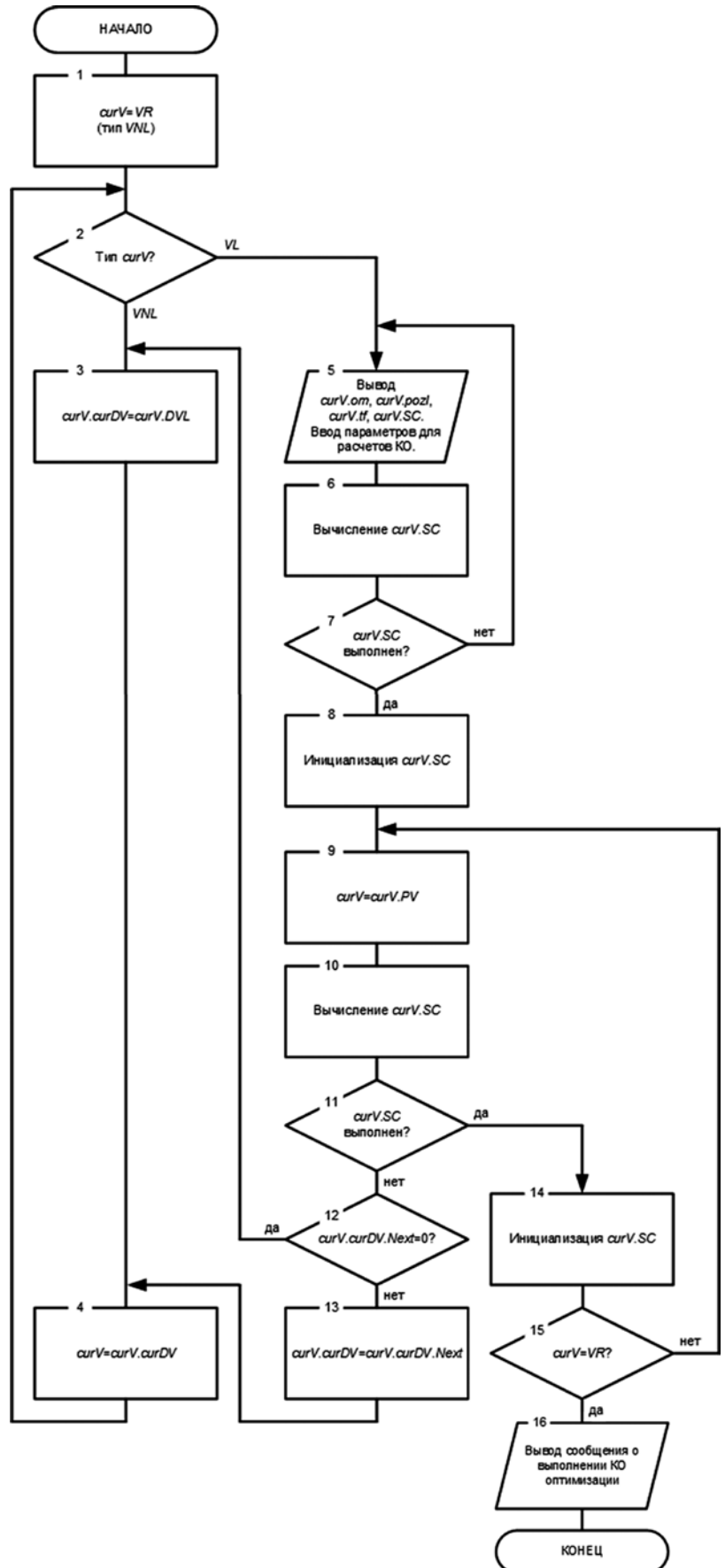


Рис. 5. Схема алгоритма обхода дерева

последним в списке потомков текущей вершины ($curV.curDV.Next = 0?$). Если да, то переходим к шагу 3, иначе переходим к шагу 13.

13. В качестве текущего потомка текущей вершины принимаем следующего потомка текущей вершины ($curV.curDV = curV.curDV.Next$). Переходим к шагу 4.

14. Инициализируем КО текущей вершины $curV.SC$, что может потребоваться поскольку КО был выполнен. Переходим к шагу 15.

15. Проверяем, является ли текущая вершина корневой в **TDMROZ**. Если да, то переходим к шагу 16, иначе переходим к шагу 9. Таким образом, если текущая нелистовая вершина $curV$ не является корневой, то текущая вершина $curV$ передаст управление родителю.

16. Выводим сообщение о выполнении КО оптимизации. Конец алгоритма обхода **TADMROZ**.

На рис. 6 приведен пример дерева ДМРОЗ.

Гибридный метод поиска

На базе ДМРОЗ и процедур ГА и ЛП-оптимизации предложим гибридный метод поиска, схема которого приведена на рис. 7. Как видно из схемы, такие шаги

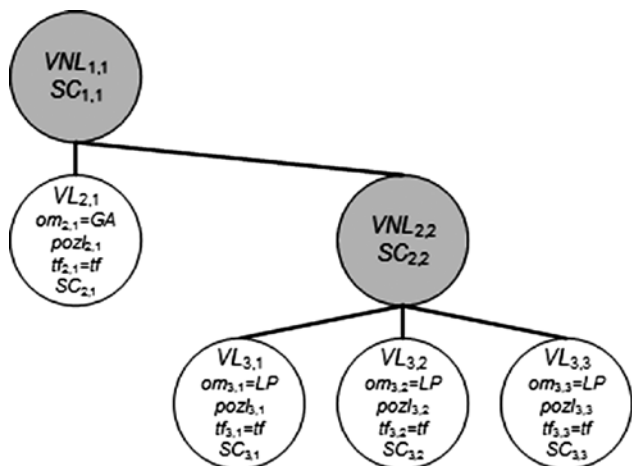


Рис. 6. Пример дерева ДМРОЗ

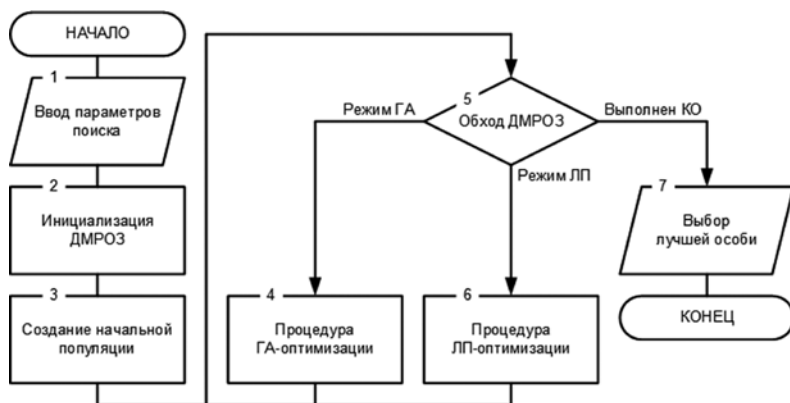


Рис. 7. Схема гибридного метода поиска

классического ГА как ввод параметров поиска, создание начальной популяции, выбор лучшей особи вынесены в схему гибридного метода поиска, а не включены в процедуру ГА-оптимизации.

Можно выделить следующие шаги гибридного метода поиска:

1. Вводим параметры поиска.
2. Инициализируем дерево ДМРОЗ.
3. Создаем начальную популяцию решений. Переходим к шагу 5.
4. Выполняем процедуру ГА-оптимизации. Переходим к шагу 5.
5. Выполняем обход дерева ДМРОЗ согласно приведенному выше алгоритму **TADMROZ**. Если в процессе обхода установлено, что КО в вершине VR выполнен, то переходим к шагу 7. Если установлена необходимость выполнения процедуры ГА-оптимизации, то переходим к шагу 4. Если установлена необходимость выполнения процедуры ЛП-оптимизации, то переходим к шагу 6.
6. Выполняем процедуру ЛП-оптимизации. Переходим к шагу 5.
7. Выбираем особь текущей популяции с наибольшим значением ЦФ. Конец.

Заключение

Предложенная в работе ДМРОЗ обеспечивает следующие возможности:

- формализацию правил сочетания ГА и методов ЛП для решения многомерных мультимодальных ОЗ, и, отдельно отметим, правил сочетания методов одномерного ЛП для выполнения многомерного ЛП;
- структурный анализ ОЗ, возможность определения оптимизационных подзадач с различными ЦФ, КО, методами оптимизации, пространствами решений;
- управление количеством времени, выделяемым на решение различных оптимизационных подзадач.

Можно предположить широкую область применения ДМРОЗ. При разработке предлагаемой модели областью применения являлся поисковый синтез алгоритма сжатия полутонновых изображений без потерь.

Литература

1. Рейзлин В.И. Численные методы оптимизации. Томск: Изд-во Томского политех. ун-та, 2011.
2. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров. М.: Высшая школа, 1994.
3. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы / под ред. В.М. Курейчика. М.: Физматлит, 2006.
4. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические

алгоритмы и нечеткие системы / пер. с польск. И.Д. Рудинского. М.: Горячая линия – Телеком, 2006.

5. **Панченко Т.В.** Генетические алгоритмы: учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. Астрахань: Издательский дом «Астраханский университет», 2007.

6. **Кормен Т., Лейзерсон Ч., Ривест Р.** Алгоритмы: построение и анализ / пер. с англ. под ред. А. Шеня. М.: МЦНМО, 2002.

7. **Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.** Алгоритмы: построение и анализ. М.: Издательский дом «Вильямс», 2005.

Статья поступила в редакцию 17.06.2013