

Мультимемеевая модификация гибридного муравьиного алгоритма непрерывной оптимизации NSIAC

09, сентябрь 2012

DOI: 10.7463/0912.0470529

Карпенко А. П., Чернобривченко К. А.

УДК 519.6

Россия, МГТУ им. Н.Э. Баумана

apkarpenko@mail.ru

chernobrivchenko.k@gmail.com

Введение

Публикация продолжает работу [1], посвященную исследованию эффективности модифицированного нами алгоритма непрерывно взаимодействующей колонии муравьев *CIAC* (*Continuous Interacting Ant Colony*) [2].

Муравьиные алгоритмы представляют собой новый класс алгоритмов оптимизации, инспирированных живой природой. В основу алгоритмов положено моделирование самоорганизации в муравьиной колонии, как многоагентной системе, в которой каждый агент (муравей) функционирует по очень простым правилам, но поведение системы в целом оказывается высоко рациональным. Муравьи в колонии образуют структуру, называемую *гетерархией*, которая, в отличие от иерархии, предполагает не вертикальные связи между агентами системы, а горизонтальные, когда все агенты равноправны между собой. В колонии муравьев имеет место *плотная гетерархия*, когда каждый из агентов системы может связываться с любым другим агентом и в любой момент времени. Общение между муравьями происходит через каналы *прямого взаимодействия* и *стигмергии* (*stigmergy*). Прямое взаимодействие между агентами осуществляется посредством

обмена пищей и выделениями желез (*трофоллаксис*). Канал стигмергии (*следовой канал*) представляет собой способ обмен информацией через феромонный след, оставляемый муравьями.

Муравьиные алгоритмы используют для решения задач, как дискретной, так и непрерывной оптимизации. Первым вариантом алгоритма колонии муравьев, разработанным для решения задачи непрерывной оптимизации, стал алгоритм непрерывной оптимизации колонией муравьев (*Continuous Ant Colony Optimization, CACO*), наметки которого были предложены Билчевым (*G. Bilchev*) и Парми (*I. C. Parmee*) в 1995 г. В настоящее время известно большое число алгоритмов непрерывной оптимизации колонией муравьев, одним из которых является упомянутый выше алгоритм *CIAC*. Для повышения эффективности алгоритма *CIAC* его авторы разработали модификацию *Hybrid CIAC (HCIAC)* [3], основной особенностью которой является реализация локального поиска с помощью широко известного алгоритма *Хелдера-Муда*.

В данной работе мы предлагаем модификацию алгоритма *HCIAC* на основе *меметических (memetic)* алгоритмов, которые представляют собой гибридные популяционные метаэвристические алгоритмы поисковой оптимизации, основанные на концепции мема и нео-дарвиновском принципе эволюции [4]. Модифицированный алгоритм назван нами *HCIAC-M*.

В контексте меметических алгоритмов мем является реализацией какого-либо алгоритма локальной оптимизации, уточняющего текущие решения на каждой итерации, либо через некоторое число итераций. В широком смысле, меметические алгоритмы можно рассматривать как гибридизацию одного из популяционных алгоритмов глобального поиска и одного или нескольких классических или популяционных алгоритмов локальной оптимизации.

Из различных вариантов меметических алгоритмов чаще всего используют вариант *мультимемных (multi-memes)* адаптивных алгоритмов. Основной задачей конструирования таких алгоритмов является выбор

целесообразной стратегии использования того или иного мема из роя доступных мемов. Задача относится к классу задач метаоптимизации [5]. Указанную стратегию иногда называют *гиперэвристикой (hyperheuristic)*.

Наиболее известны три следующие категории гиперэвристик, используемых в *гиперэвристических адаптивных мультимемных алгоритмах*: случайные (*random*) гиперэвристики, жадные (*greedy*) гиперэвристики, гиперэвристики с функцией выбора (*choice-function*). В алгоритме *НСИАС-М* использован один из вариантов жадной гиперэвристики.

НСИАС-М включает в себя еще одну модификацию алгоритма *НСИАС* – последний алгоритм дополнен процедурой *встряски популяции*, которая призвана диверсифицировать поиск и тем самым повысить вероятность локализации глобального экстремума целевой функции.

В пункте 1 приведена постановка задачи и представлены схемы алгоритмов *СИАС* и *НСИАС*. Предлагаемая модификация алгоритма *НСИАС* рассматривается в пункте 2. Пункт 3 посвящен исследованию эффективности алгоритма *НСИАС-М* при поиске глобального экстремума одноэкстремальной овражной функции Розенброка (*Rosenbrock*) и многоэкстремальной функции Растригина (*Rastrigin*) из известного пакета тестовых функций *CES (Congress of Evolutionary Computing)* [6]. Последовательно представлены результаты исследования модифицированных встряской алгоритмов *СИАС* и *НСИАС*, а также алгоритма *НСИАС-М*. В пункте 4 в качестве примера применения алгоритма *НСИАС-М* рассмотрена практически значимая задача оптимизации системы двигатель-генератор [7], сводящаяся к задаче оптимального управления, которую мы решаем путем редукции к задаче нелинейного программирования, а затем к задаче безусловной оптимизации.

1. Постановка задачи и схема базового муравьиного алгоритма

Рассматриваем задачу многомерной глобальной безусловной оптимизации: найти максимум целевой функции $f(X)$, определенной на множестве D евклидова пространства $R^{|X|}$:

$$\max_{X \in D \subset R^{|X|}} f(X) = f(X^*) = f^*. \quad (1)$$

Здесь $X = (x_1, x_2, \dots, x_{|X|})$ - вектор варьируемых параметров,

$$D = \{X \mid x_i^- \leq x_i \leq x_i^+, i \in [1:|X|]\} - \quad (2)$$

параллелепипед допустимых значений этого вектора.

Как мы отмечали выше, алгоритм *НСИАС* построен на основе алгоритма *СИАС*. Рассмотрим поэтому прежде последний алгоритм [2].

1.1. Алгоритм СИАС

Схема алгоритма *СИАС* имеет следующий вид.

1) Создаем агентов (муравьев) s_j , $j \in [1:|S|]$ и случайным образом распределяем их в области поиска D . Здесь $|S|$ - число агентов в колонии S .

2) Моделируем работу канала стигмергии, для чего для каждого из агентов s_j вычисляем координаты центра тяжести феромонных точек

$$\Phi_j = \sum_{i=1}^{|\Phi|} \bar{\omega}_{i,j} X_i,$$

где X_i - вектор координат i -ой феромонной точки, $|\Phi|$ - их общее число,

$$\bar{\omega}_{i,j} = \frac{\omega_{i,j}}{\sum_{i=1}^{|\Phi|} \omega_{i,j}} -$$

нормированный *интерес* j -го агента к i -ой феромонной точке;

$$\omega_{i,j} = \frac{\bar{\delta}}{2} \theta_i \exp(-\delta_{i,j}).$$

Здесь $\bar{\delta}$ - среднее расстояние между двумя агентами в колонии, θ_i - количество феромона в точке X_i , $\delta_{i,j}$ - расстояние между j -м агентом и i -ой феромонной точкой.

3) Перемещаем агента s_j в направлении точки Φ_j на случайное расстояние в пределах его *размерного параметра* $r_j = \frac{\xi_r}{\psi}$, где ξ_r – случайная величина, распределенная по нормальному закону с нулевым математическим ожиданием и среднеквадратичным отклонением σ ; ψ – делитель *размерного параметра*, определяющий величину этого перемещения.

4) Моделируем функционирование канала прямого взаимодействия – агент s_j читает из своей памяти случайное сообщение, полученное от агента s_k .

5) Сравниваем значение целевой функции $f(X_j)$ в точке расположения агента s_j с соответствующим значением $f(X_k)$, полученным в сообщении от агента s_k . Если $f(X_k) > f(X_j)$, то агента s_j перемещаем в случайную точку гиперсферы, центром которой является точка X_k , а радиусом – размерный параметр r_j этого агента.

6) Если $f(X_k) \leq f(X_j)$, то агент s_j посылает сообщение со своими координатами X_j и значением целевой функции $f(X_j)$ случайному агенту популяции, а затем в пределах своего размерного параметра r_j совершает случайное перемещение в окрестности точки X_j .

7) После того, как по рассмотренной схеме все агенты колонии завершили свои перемещения, моделируем испарение феромонных точек (см. ниже).

8) Проверяем выполнение критерия останова, в качестве которого может использоваться условие достижения заданного числа итераций $t = \hat{t}$ или условие стагнации вычислительного процесса

$$f(X(t)) - f(X(t - t_\delta)) \leq \varepsilon_f,$$

где t_δ – заданное число итераций, ε_f – малая положительная величина.

В каждой точке $X_j(\tau)$, в которой побывал муравей s_j , он оставляет феромонную точку $\phi_j(t, \tau)$, интенсивность которой (количество феромона) на данной итерации t равна $\theta_j(t, \tau)$, где $\tau \in [0:t]$ - номер итерации, на которой оставлена эта точка. Количество феромона в феромонных точках с ростом числа итераций уменьшается (феромон испаряется) в соответствии с рекуррентной формулой

$$\theta_j(t, \tau) = \begin{cases} \rho \theta_j(t-1, \tau), & \theta_j(t-1, \tau) \geq \theta_{\min}, \\ 0, & \text{иначе,} \end{cases} \quad (3)$$

где $\rho \in (0; 1)$ – коэффициент устойчивости феромонных точек. Формула означает, что феромонная точка исчезает, если количество феромона в ней становится равным величине θ_{\min} . Начальное количество феромона агента s_1 полагаем равным единице, а остальных агентов – пропорциональным соответствующим значениям целевой функции.

Размер памяти агента ограничен, так что максимально допустимое число сообщений в памяти каждого из агентов не превышает величины μ_{\max} . В ситуации, когда возможно переполнение памяти (агент получает новое сообщение при текущем числе сообщений в его памяти, равном μ_{\max}), одно из выбранных случайным образом сообщений удаляем из памяти, а новое сообщение записываем в нее.

1.2. Алгоритм *НСИАС*

Алгоритм *НСИАС* использует следующие модификации алгоритма *СИАС* [3]. Во-первых, как мы отмечали выше, *НСИАС* реализует локальный поиск на основе алгоритма Нелдера-Мида. Во-вторых, *НСИАС* использует модифицированную модель канала стигмергии.

В алгоритме *СИАС* в случае медленного испарения феромонных точек число этих точек может стать чрезвычайно большим, что приведет к значительному увеличению требуемой памяти ЭВМ и высоким вычислительным затратам. В алгоритме *НСИАС* для решения этой проблемы

муравьи могут не только добавлять новые точки, но и усиливать старые, находящиеся в *зоне видимости* (*visible zone*) агентов.

Схему алгоритма *НСИАС* передает следующая последовательность его основных шагов.

1) Инициализируем агентов по схеме алгоритма *СИАС*. Кроме того задаем начальные радиусы зон видимости d_j , $j \in [1:|S|]$, подчиненные нормальному закону распределения с математическим ожиданием и среднеквадратичным отклонением равными соответственно m_d , σ_d .

2) Моделируем испарение феромонных точек по формуле (3).

3) Для каждого из агентов s_j реализуем этап глобального поиска.

3.1) Моделируем выбор агентом s_j канала стигмергии или канала прямого взаимодействия на основании *функции выбора* (*stimulus-response function*) вида

$$\xi(e) = \frac{1}{1 + \exp(-b_s e + b_h)}, \quad (4)$$

где $\xi(e) \in [0; 1]$ - вероятность выбора канала стигмергии, e – *привлекательность* (*stimulus*) канала; b_s , b_h - одинаковые для всех агентов *коэффициент гладкости* (*smoothness*) и величина *порога чувствительности* (*threshold*) соответственно.

3.2) Если для агента s_j был выбран канал стигмергии, то отыскиваем все феромонные точки в пределах его зоны видимости d_j и по схеме алгоритма *СИАС* реализуем движение агента в направлении их центра тяжести.

3.3) Если для агента s_j выбрал канал прямого взаимодействия, то извлекаем из памяти этого агента первое сообщение и обрабатываем его в соответствии с алгоритмом *СИАС*.

4) Если в зоне видимости агента s_j нет феромонных точек (случай 3.2) или в его памяти нет сообщений (случай 3.3), то основываясь на

мотивации (*motivation*) агента λ_j по формуле вида (4) производим выбор в пользу работы агента s_j с каналом прямого взаимодействия или в пользу локального поиска. Здесь величина λ_j в отличие от привлекательности e_j , является динамической величиной с нулевым начальным значением.

4.1) Если выбор сделан в пользу работы агента s_j с каналом прямого взаимодействия, то моделируем отправку этим агентом сообщения случайному агенту популяции и полагаем $\lambda_j = \lambda_j + \delta_\lambda$, то есть, увеличиваем мотивацию агента на небольшую величину $\delta_\lambda > 0$.

4.2) В случае выбора второго варианта действий, исходя из текущего положения агента X_j , полагаем мотивацию λ_j равной нулю и выполняем следующие действия.

4.2.1) Запускаем локальный поиск алгоритмом Нелдера-Мида, полагая начальную длину ребра симплекса, равной текущему радиусу зоны видимости d_j . Локальный поиск ограничиваем n_f^{\max} оценками значений целевой функции. В качестве конечного положения агента s_j принимаем положение, соответствующее лучшему достигнутому значению функции $f(X)$.

4.2.2) Отыскиваем видимые данным агентом феромонные точки.

4.2.3). Если указанные точки имеются, усиливаем ближайшую из них X_k в соответствии с формулой

$$\theta_k = \theta_k + \frac{\theta_k}{2},$$

а радиус видимости d_j уменьшаем до величины, равной расстоянию данного агента до наиболее дальней феромонной точки в зоне его видимости.

4.2.4) Если указанные феромонные точки отсутствуют, формируем новую феромонную точку X_j с количеством феромона, пропорциональным $f(X_j)$. Мотивацию λ_j устанавливаем равной нулю.

5) Реализуем случайное перемещение агента s_j в пределах его текущей зоны видимости d_j .

6) По правилам алгоритма *CIAS* продолжаем либо завершаем итерации.

2. Модификация базового алгоритма

Результаты исследований, выполненных авторами алгоритма *NCIAS*, показывают, что этот алгоритм по сравнению с алгоритмом *CIAS* позволяет значительно повысить скорость сходимости итераций. Однако ценой этого является значительное увеличение числа испытаний (вычислений значений целевой функции). Так, в наших экспериментах показано, что один акт локального поиска пятимерной функции Розенброка может требовать до 800 испытаний. Причиной этого является низкая скорость сходимости алгоритма Нелдера-Мида на некоторых классах функций. При решении практически значимых задач оптимизации, в которых целевая функция имеет высокую вычислительную сложность, данное обстоятельство может привести к недопустимо большим вычислительным затратам.

В алгоритме *NCIAS-M* в качестве мемов используем алгоритм m_1 Нелдера-Мида, алгоритм m_2 ортогонального исследования области поиска [8], квазиньютоновский алгоритм m_3 со смешанной процедурой квадратичного и кубического поиска [9]. Таким образом, рой мемов состоит из мемов m_1, m_2, m_3 и число мемов $|M| = 3$.

Используем жадную гиперэвристику, основные особенности которой заключаются в следующем.

1) Лучшего для данного агента $s_j, j \in [1:|S|]$ мема выбираем на основе h запусков локального поиска.

2) Критерий эффективности мема m_k строим на основе среднего числа испытаний за один запуск алгоритма локального поиска

$$\bar{n}_f(m_k) = \frac{1}{h} \sum_{p=1}^h n_{f,p}(m_k), \quad k \in [1:3],$$

где $n_{f,p}(m_k)$ - число испытаний в процессе p -го локального поиска мемом m_k . Точнее говоря, мема m_k объявляем победителем, если

$$\bar{n}_f(m_k) = \min_{l \in [1:3]} \bar{n}_f(m_l)$$

и если лучшее решение $f(X_k^*)$, полученное им, не более чем на величину $\varepsilon_f > 0$ меньше лучшего решения $f(X_l^*)$, найденного популяцией мемов, то есть

$$f(X_l^*) - f(X_k^*) \leq \varepsilon_f, \quad l \in [1:3], \quad l \neq k.$$

3) На последних t_f итерациях мема-победителя выбираем только по второму из указанных критериев, то есть полагаем, что

$$\max(f(X_1^*), f(X_2^*), f(X_3^*)) = f(X_k^*).$$

Как отмечалось выше, алгоритм *НСИАС-М* включает в себя еще процедуру встряски популяции, которая запускается с шагом

$$t_\Delta = t_\Delta^0 + \alpha t,$$

где t - текущий номер итерации; t_Δ^0, α - свободные параметры алгоритма. В соответствии с этой формулой с ростом числа итераций шаг встряски t_Δ увеличивается, обеспечивая переход от исследования пространства поиска к точной локализации найденного максимума целевой функции.

Суть процедуры встряски популяции заключается в следующем.

1) После завершения очередных t_Δ итераций отбираем $|\tilde{S}|$ не лучших агентов \tilde{s}_{j_k} с координатами \tilde{X}_{j_k} и распределяем их случайным образом в области поиска в точках \tilde{X}'_{j_k} соответственно; $j_k \in [1:|\tilde{S}|], k \in [1:|\tilde{S}|]$.

2) Вычисляем значения целевой функции в точках \tilde{X}'_{j_k} и создаем в этих точках феромонные пятна, интенсивность которых полагаем равной

$$\tilde{\theta}'_{jk} = \beta \tilde{\theta}_{jk} \frac{f(\tilde{X}_{jk})}{f(\tilde{X}'_{jk})}, \quad (5)$$

где $\tilde{\theta}_{jk}$ - количество феромона в точке \tilde{X}_{jk} , $\beta \geq 1$ - свободный параметр.

3. Исследование эффективности алгоритма *НСИАС-М*

Модификация алгоритма *НСИАС* была выполнена в несколько этапов. Соответственно многоэтапным было исследование его эффективности.

Реализация рассматриваемых алгоритмов выполнена среде *MatLab*. Вычислительные эксперименты проводились на персональном компьютере с процессором *Intel Core 2 Duo T8100* 2,1 ГГц и оперативной памятью объемом 2 Гб.

3.1. Модификация алгоритма *СИАС* встряской.

В качестве тестовой использована пятимерная функция Растригина. Если не указано иное, вычислительные эксперименты выполнены со значениями свободных параметров модифицированного алгоритма *СИАС*, называемого нами *СИАС-S*, которые использованы в работе [1]: коэффициент устойчивости феромонных точек $\rho = 0,9$; минимальное количество феромона $\theta_{\min} = 0,00000001$; среднеквадратичное отклонение размерного параметра $\sigma = 0,1$; число агентов в колонии $|S| = 186$; делитель размерного параметра $\psi = 50$; максимально допустимое число итераций $\hat{t} = 5000$; период стагнации $t_{\delta} = 1500$. Число стартов алгоритма принято равным 30; $\alpha = 0,01$; $\beta = 1$; $t_{\Delta}^0 = 1$; $\varepsilon_f = 10^{-3}$. Относительное число агентов, подвергшихся перемещению

в процессе встряски популяции, равно $\eta = \frac{|\tilde{S}|}{|S|} = 0,99$.

В качестве критерия эффективности алгоритма используем величину ζ_{\sim} - оценку вероятности локализации глобального максимума функции.

Данная оценка во всех случаях получена по итогам мультистарта как процент успешных стартов, то есть стартов, в результате которых глобальный максимум найден с требуемой точностью.

Рисунок 1 иллюстрирует вероятность $\tilde{\xi}$ в функции величины делителя размерного параметра ψ . Из сравнения этого рисунка с рисунком 13 публикации [1] вытекает, что встряска обеспечивает значительное повышение вероятности локализации глобального экстремума. Так, при $\psi = 4$ алгоритм *CIAC-S* обеспечивает $\tilde{\xi} = 0,98$, в то время как исходный алгоритм – всего $\tilde{\xi} = 0,6$. В дальнейших экспериментах принято $\psi = 10$.

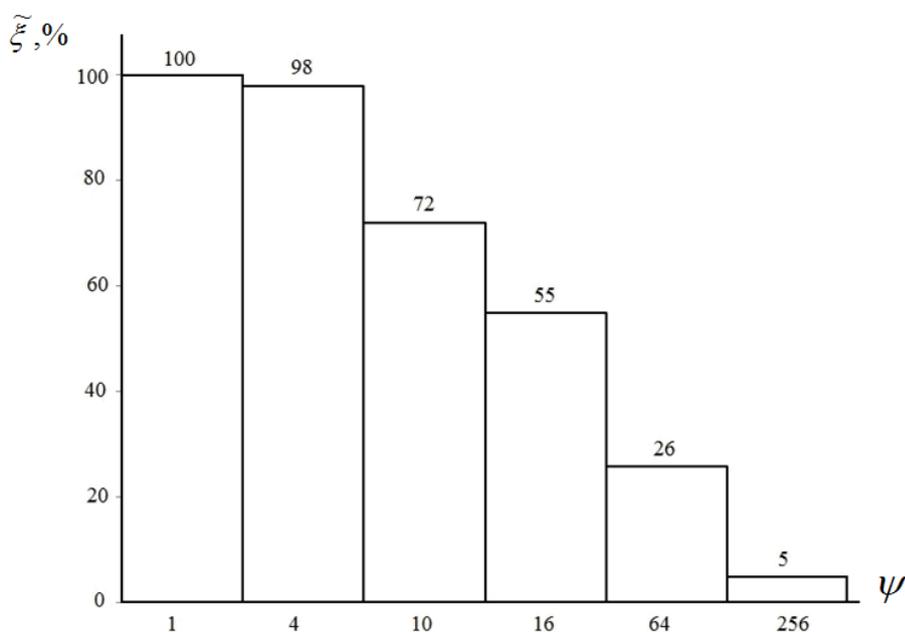


Рисунок 1 – Оценка вероятности $\tilde{\xi}$ в функции значений параметра ψ : пятимерная функция Растригина; алгоритм *CIAC-S*

Зависимость вероятности $\tilde{\xi}$ от значений параметра α , отвечающего за частоту запуска процедуры встряски, показывает рисунок 2.

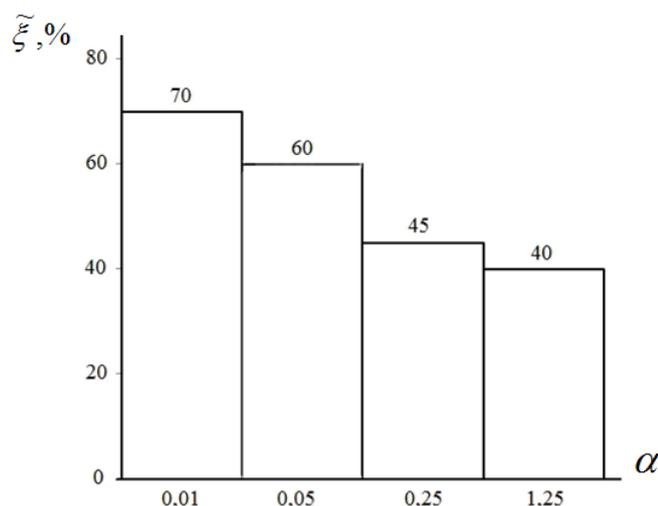


Рисунок 2 – Оценка вероятности $\tilde{\xi}$ в функции значений параметра α :
пятимерная функция Растригина; алгоритм *CIAC-S*

Из рисунка 2 следует, что, как и ожидалось, малые значения величины α , которым соответствует высокая частота встряски, приводят к более высокой вероятности локализации глобального экстремума. Ценой этого является увеличение числа испытаний и, как следствие, времени решения задачи. Поэтому для дальнейших исследований принимаем компромиссное значение параметра α , равное 0,025.

Результаты исследования зависимости $\tilde{\xi}(\beta)$ иллюстрирует рисунок 3. Заметим, что, как следует из формулы (5), большим значениям величины β соответствует большее количество феромона, откладываемого в новых точках, то есть в точках, возникших в результате реализации процедуры встряски.

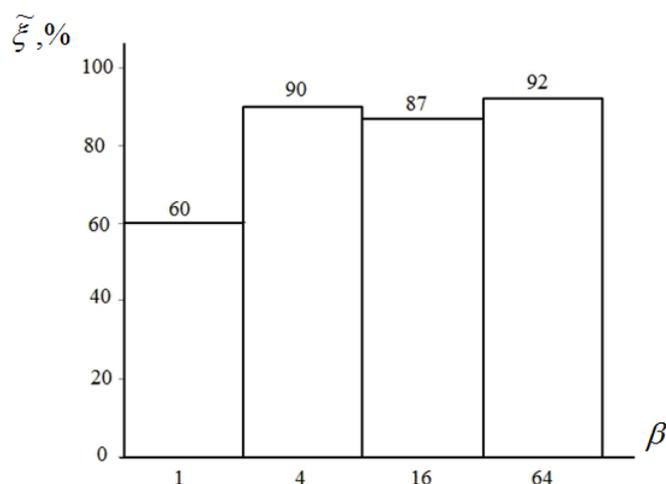


Рисунок 3 – Оценка вероятности $\xĩ$ в функции значений параметра β :
 пятимерная функция Растригина; алгоритм *CIAC-S*

Рисунок 3 показывает, что увеличение значений параметра β ожидаемо повышает вероятность локализации экстремума. Однако при этом значительно возрастает число испытаний, поскольку из-за большего количества феромона в новых точках он в этих точках испаряется медленнее. На основе данного исследования для многоэкстремальных функций типа функции Растригина можно рекомендовать значение $\beta = 4$, которое мы используем в последующих экспериментах.

Результаты варьирования относительного числа агентов η , участвующих в процедуре встряски, иллюстрирует рисунок 4.

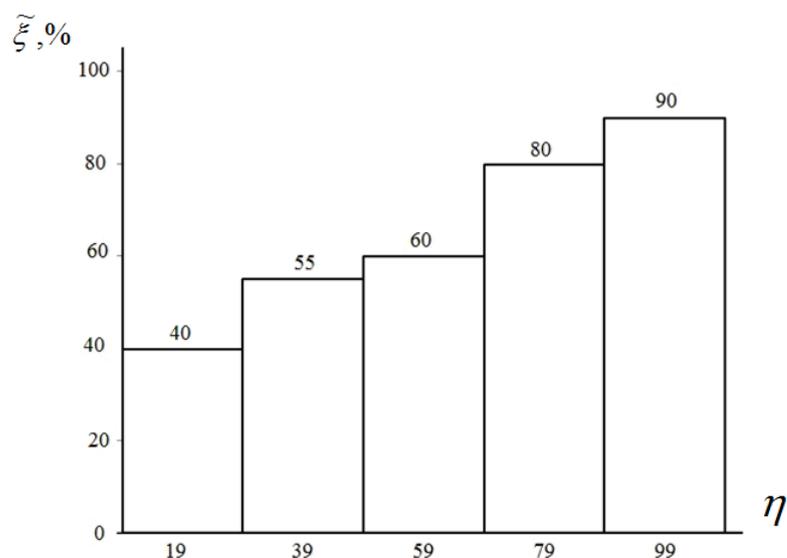


Рисунок 4 – Оценка вероятности ξ в функции значений параметра η :
 пятимерная функция Растригина; алгоритм *CIAC-S*

Из рисунка 4 отчетливо видно, что встряска популяции повышает вероятность локализации глобального экстремума в тем большей степени, чем выше значение этого параметра. Однако с ростом величины η быстро растет требуемое число испытаний, так что в качестве оптимального значения этой величины мы рекомендуем значение $\eta = 0,79$, используемое в дальнейших вычислительных экспериментах.

Таким образом, представленные результаты показывают эффективность предложенной модификации алгоритма *CIAC*. Исследование позволило определить оптимальные для пятимерной функции Растригина значения свободных параметров процедуры встряски. Есть основания полагать, что эти значения будут близки к оптимальным и для других многоэкстремальных функций. Важно, что процедура встряски не уменьшает результирующую точность алгоритма, поскольку на его завершающих итерациях эта процедура «отключается». Процедура оставляет широкие возможности оптимизации ее свободных параметров по критериям вероятности локализации глобального экстремума и вычислительной сложности.

3.2. Модификация алгоритма *HCIAC* встряской

Исследование эффективности данной модификации алгоритма *HCIAC*, названной нами *HCIAC-S*, выполнено на функциях Розенброка и Растригина. В качестве значений свободных параметров алгоритма использованы значения, указанные в п. 3.1.

Зона видимости агента в алгоритме *HCIAC* динамически меняется в процессе поиска, поэтому исследование зависимости $\tilde{\xi}(\psi)$ в данном случае не имеет смысла. В качестве основного варьируемого параметра рассматриваем число агентов $|S|$ в популяции.

Исследование выполнено для пяти- и десятимерной функций Розенброка. В работе [3] показано, что при поиске экстремума этой функции с помощью канонического алгоритма *HCIAC*, возникает ряд проблем. Модификация алгоритма *HCIAC* встряской имела целью, как отмечалось выше, повышение эффективности алгоритма при поиске экстремума многоэкстремальных функций. Использование в качестве тестовых функций Розенброка разной размерности позволяет проверить насколько процедура встряски эффективна для иного класса функций – класса овражных функций.

Результаты исследования для пятимерной функции Розенброка иллюстрирует рисунок 5, а для десятимерной функции – рисунок 6.

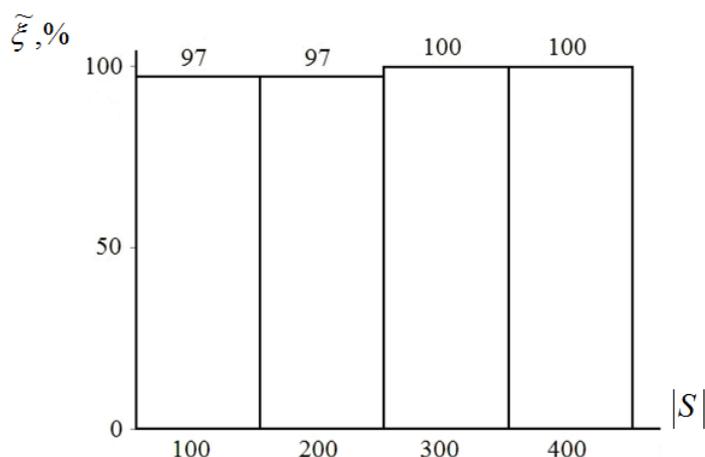


Рисунок 5 – Оценка вероятности $\tilde{\xi}$ в функции размера популяции $|S|$: пятимерная функция Розенброка; алгоритм *HCIAC-S*

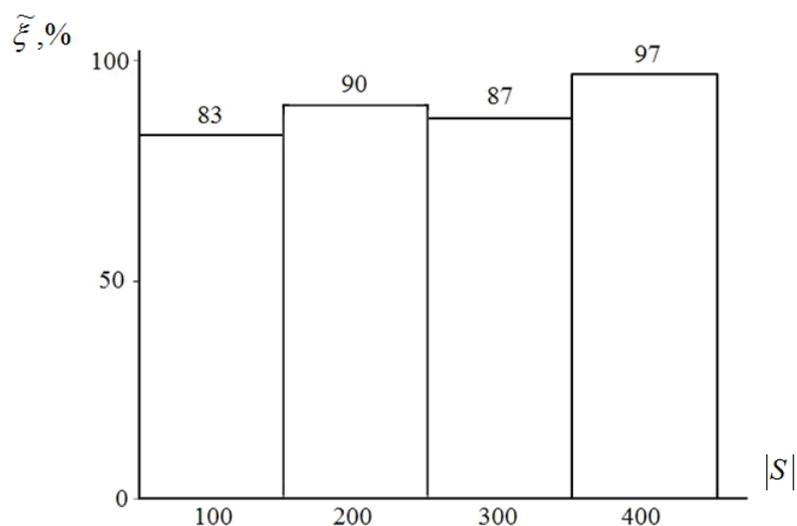


Рисунок 6 – Оценка вероятности $\xi_{\tilde{}}$ в функции размера популяции $|S|$: десятимерная функция Розенброка; алгоритм *HCIAC-S*

Рисунок 5 показывает, что популяция уже из 100 агентов обеспечивает практически стопроцентную вероятность локализации экстремума пятимерной функции Розенброка. Поскольку увеличение числа агентов популяции вызывает почти пропорциональный рост требуемого числа испытаний (в рассматриваемом примере с $\sim 9,9 \cdot 10^5$ испытаний при $|S|=100$ до $\sim 4 \cdot 10^6$ при $|S|=400$), такое поведение величины $\xi_{\tilde{}}$ является важным с точки зрения сокращения вычислительных затрат. Аналогичные выводы можно сделать из рисунка 6 для десятимерной функции Розенброка. Заметим, что в тех же условиях для обеспечения 100 % вероятности локализации экстремума алгоритм *CIAC* требовал число агентов в популяции, равное $|S|=398$ [1].

Отметим следующее обстоятельство. Благодаря динамическому изменению зоны видимости агента и гибридизации с алгоритмом локального поиска, алгоритм *HCIAC-S* в обоих экспериментах обеспечил среднюю абсолютную погрешность порядка 10^{-8} - 10^{-9} , что на порядок меньше лучшего результата алгоритма *CIAC*.

Некоторые результаты исследования эффективности алгоритма *HCIAC-S* для функции Растригина представлены на рисунке 7.

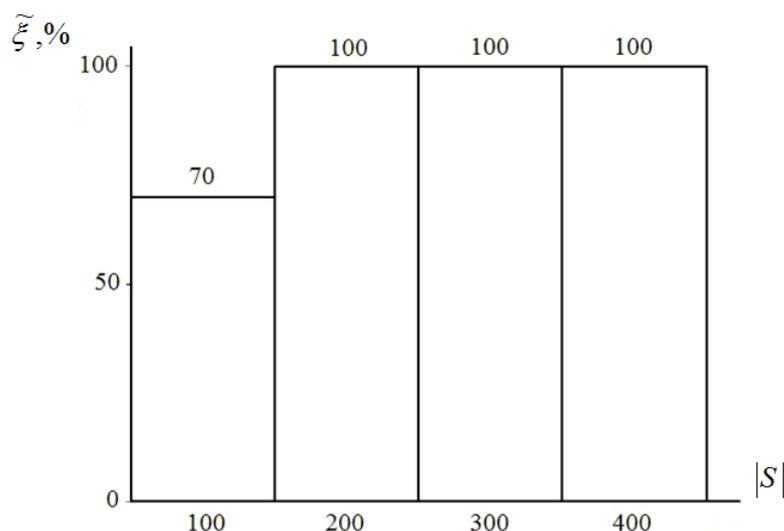


Рисунок 7 – Оценка вероятности $\tilde{\xi}$ в функции размера популяции $|S|$: пятимерная функция Розенброка; алгоритм *HCIAC-S*

Из рисунка 7 вытекает, что алгоритм *HCIAC* обеспечивает 100 % вероятность локализации глобального экстремума пятимерной функции Растригина уже при 200 агентах (против 398 агентов алгоритма *CIAC* [1]). Данное обстоятельство, как отмечалось выше, позволяет значительно уменьшить вычислительные затраты. Алгоритм обеспечил среднюю абсолютную погрешность решения порядка $10^{-7} - 10^{-8}$, то есть на два-три порядка меньше, чем лучший результат алгоритма *CIAC*.

Итого, на основе представленных результатов можно сделать вывод, что гибридизация алгоритма *CIAC* с алгоритмом локального поиска Нелдера-Мида, а также использование процедуры встряски позволяют существенно повысить эффективность алгоритма, как по критерию вероятности локализации глобального экстремума, так и по критериям средней абсолютной погрешности решения и суммарных вычислительных затрат. При этом снижение вычислительных затрат имеет место, не смотря на то, что применение алгоритма Нелдера-Мида в локальном поиске неизбежно влечет

за собой увеличение суммарного числа испытаний, необходимых для эволюции каждого из агентов в течение одной итерации. Эффект обусловлен следующими причинами: возможность использования меньшего числа агентов для обеспечения той же точности поиска; уменьшение общего числа феромонных точек в зоне поиска; более высокая скорость сходимости алгоритма.

3.3. Мультимемеевая модификация алгоритма *HCIAC*

Эффективность алгоритма *HCIAC-M* исследуем на пятимерных тестовых функциях Розенброка и Растригина. Мультимемеевую модификацию в экспериментах, результаты которых представлены ниже, определяют следующие значения свободных параметров: число запусков локального поиска $h = 200$, точность решения $\varepsilon_f = 10^{-3}$, число последних итераций $t_f = 3$. Остальные свободные параметры алгоритма *HCIAC-M* имеют значения, указанные в пп. 3.1, 3.2.

Эффективность рассматриваемых мемов для функций Розенброка и Растригина иллюстрируют рисунки 8, 9 соответственно, на которых величина $\tilde{\chi}_k$ представляет собой оценку вероятности использования мема m_k ; $k = 1, 2, 3$. Напомним, что мему m_1 соответствует алгоритм Нелдера-Мида, мему m_2 - алгоритм ортогонального исследования области поиска, мему m_3 - квазиньютоновский алгоритм со смешанной процедурой квадратичного и кубического поиска (п. 2).

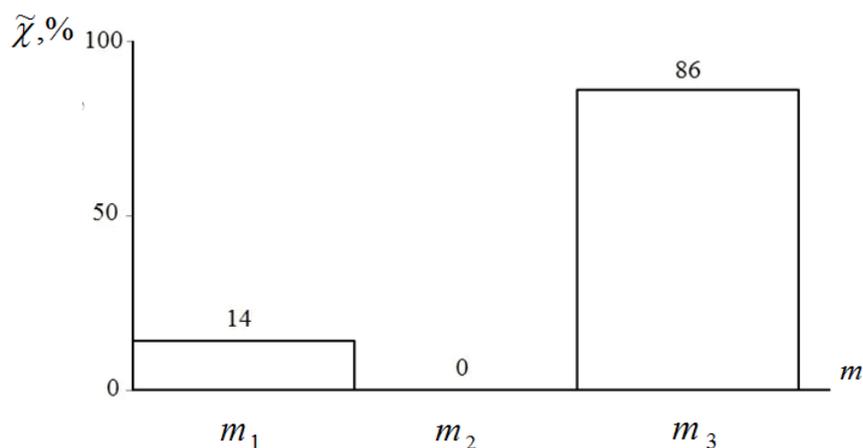


Рисунок 8 – Оценка вероятности использования мемов: функция Розенброка ($|S| = 5$); алгоритм *НСИАС-М*

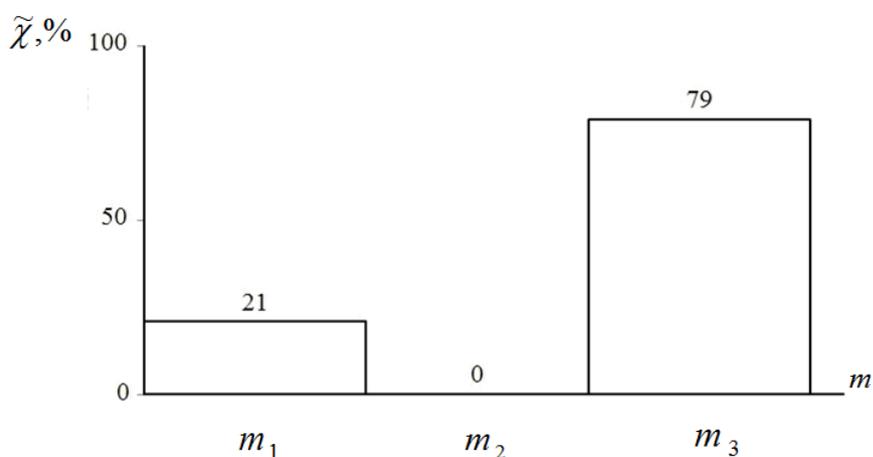


Рисунок 9 – Оценка вероятности использования мемов: функция Растригина ($|S| = 5$); алгоритм *НСИАС-М*

Рисунки 8, 9 получены при использовании в качестве меры эффективности мемов среднего числа испытаний \bar{n}_f за один запуск алгоритма локального поиска. Рисунки показывают, что в 86 % и 79 % случаев для функции Розенброка и Растригина соответственно, лучший результат показал квазиньютоновский алгоритм. Аналогично в 14 % и в 21 % случаев лучшие результаты продемонстрировал алгоритм Нелдера-Мида. Алгоритм ортогонального поиска по данному критерию не смог составить конкуренцию квазиньютоновскому алгоритму и алгоритму Нелдера-Мида. Отметим слабую чувствительность представленных результатов к классу экстремизируемой функции.

На рисунках 10, 11 представлены средние по числу запусков локального поиска погрешности $\bar{\varepsilon}$ мемов m_1, m_3 . Аналогичные результаты для ортогонального алгоритма не приведены, поскольку его эффективность по данному критерию оказалась не порядок ниже.

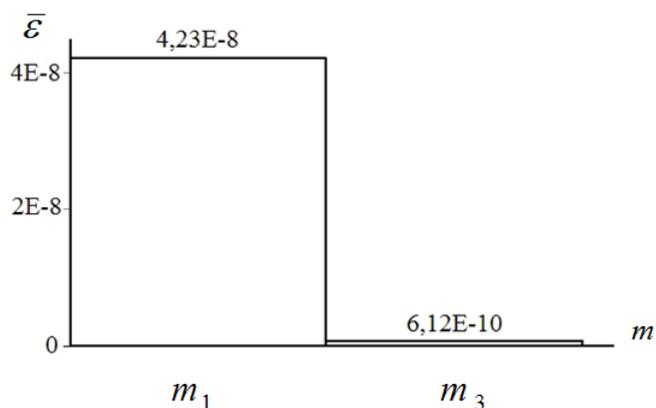


Рисунок 10 – Средняя погрешность локального поиска мемами m_1, m_3 : функция Розенброка ($|S|=5$); алгоритм *HCIAC-M*

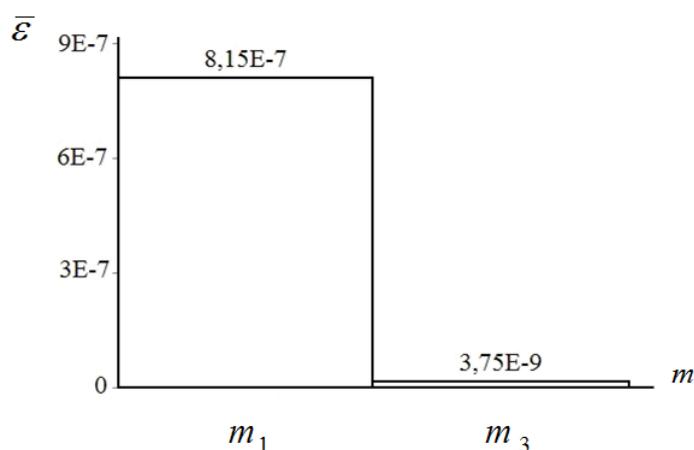


Рисунок 11 – Средняя погрешность локального поиска мемами m_1, m_3 : функция Растригина ($|S|=5$); алгоритм *HCIAC-M*

Из рисунков 10, 11 следует, что средняя точность квазиньютоновского алгоритма примерно на два порядка выше такой же точности алгоритма Нелдера-Мида.

Средние числа испытаний \bar{n}_f за один запуск алгоритма локального поиска для мемов m_1, m_2, m_3 иллюстрируют рисунки 12, 13.

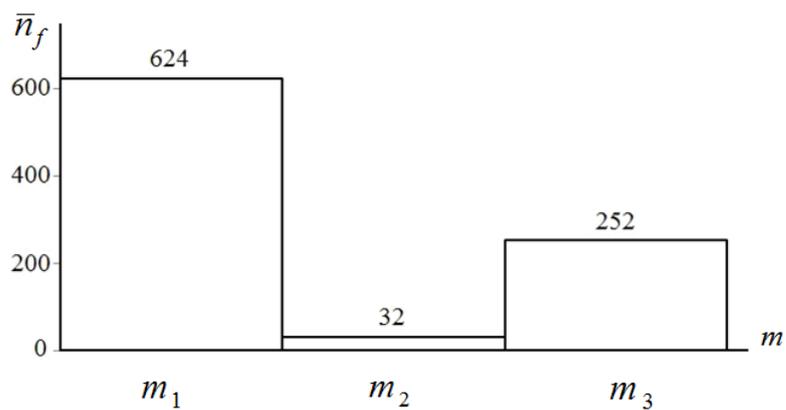


Рисунок 12 – Средняя число испытаний за один запуск алгоритма локального поиска для мемов m_1, m_2, m_3 : функция Розенброка ($|S|=5$); алгоритм *НСИАС-М*

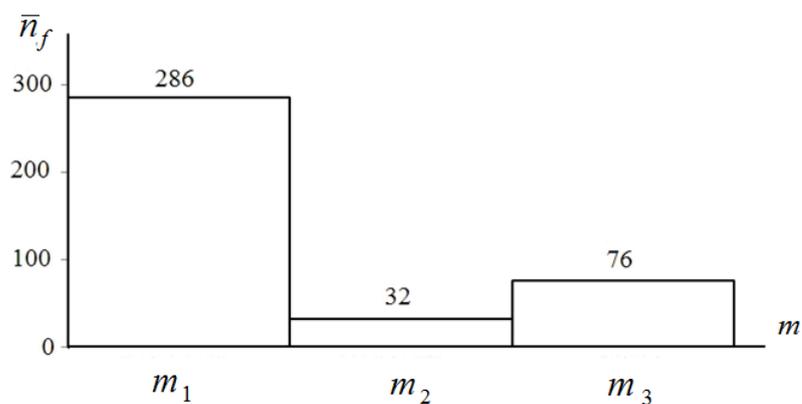


Рисунок 13 – Средняя число испытаний за один запуск алгоритма локального поиска для мемов m_1, m_2, m_3 : функция Растригина ($|S|=5$); алгоритм *НСИАС-М*

Рисунки 12, 13 показывают, что по критерию \bar{n}_f в несколько раз лучшим является алгоритм ортогонального поиска. По этому же критерию преимущество квазиньютоновского алгоритма над алгоритмом Нелдера-Мида достигает почти четырех раз (для функции Растригина).

Представленные результаты исследования свидетельствуют о том, что задачу выбора лучшего мема следует рассматривать как трех- или более критериальную задачу.

3.4. Сравнение эффективности алгоритмов *CIAC*, *HCIAC* и *HCIAC-M*

Введем в рассмотрение еще один критерий эффективности алгоритма - среднее по мультистарту число испытаний \bar{n}_f^Σ за время работы алгоритма. Значения этого критерия для функции Растригина ($|S|=5$) и алгоритмов *CIAC*, *HCIAC* и *HCIAC-M* представлены на рисунке 14. Используются указанные выше значения свободных параметров алгоритмов. Число агентов в популяции $|S|$ для алгоритма *CIAC* принято равным 398, а для алгоритмов *HCIAC*, *HCIAC-M* – равным 400.

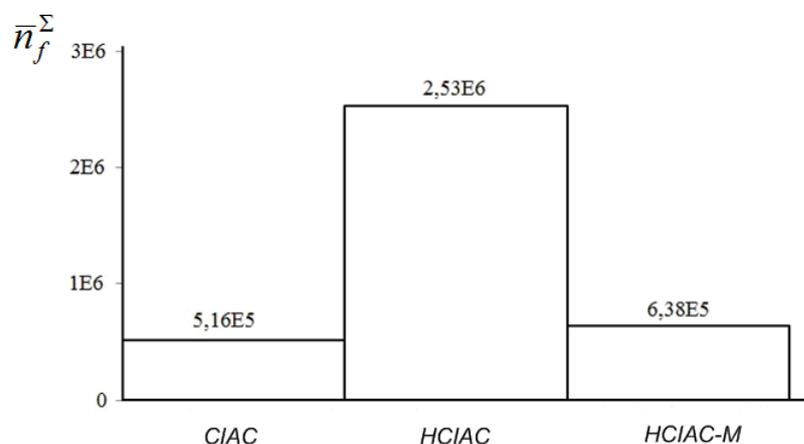


Рисунок 14 – Среднее по мультистарту число испытаний за все время решения задачи: функция Растригина ($|S|=5$); алгоритмы *CIAC*, *HCIAC*, *HCIAC-M*

Из рисунка 14 вытекает, что алгоритм *HCIAC-M* обеспечивает почти тоже суммарное число испытаний, что алгоритм *CIAC* и почти в четыре раза меньшее число, чем алгоритм *HCIAC*. В целом, результаты данного исследования показывают, что алгоритм *HCIAC-M* позволяет для пятимерной функции Растригина примерно в четыре раза повысить быстродействие, сохранив практически прежнюю вероятность локализации глобального экстремума.

4. Пример

Рассматриваем силовую часть электрического привода типа *генератор-двигатель*, предназначенного для управления некоторым рабочим механизмом (рисунок 15) [7].

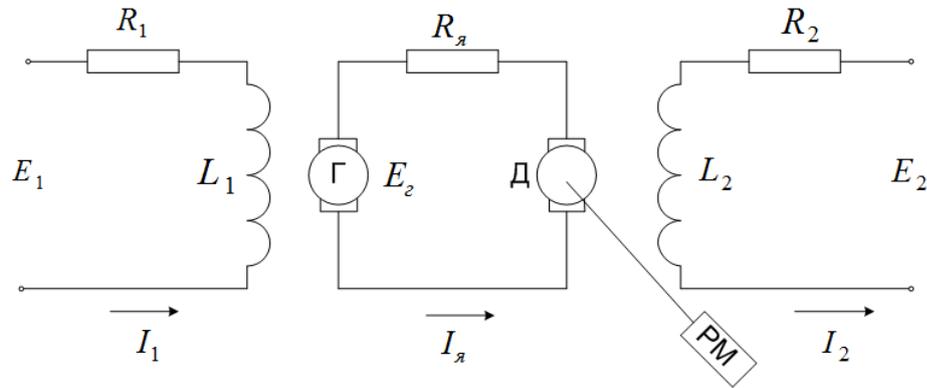


Рисунок 15 – Схема силовой части электрического привода типа генератор-двигатель: Г – генератор; Д – двигатель; PM – рабочий механизм

Динамику системы описываем системой алгебро-дифференциальных уравнений вида

$$\begin{cases} J \ddot{\psi} = M_{\delta} - M_c, & \psi(0) = \psi^0, \quad \dot{\psi}(0) = \dot{\psi}^0, \\ L_1 \dot{I}_1 = E_1 - I_1 R_1, & I_1(0) = I_1^0, \\ L_2 \dot{I}_2 = E_2 - I_2 R_2, & I_2(0) = I_2^0, \end{cases} \quad (6)$$

$$I_{\text{я}} = \frac{\Phi_2(I_1) - c I_2 \dot{\psi}}{R_{\text{я}}}, \quad (7)$$

где приняты следующие обозначения: J — момент инерции якоря двигателя и приводимого им в движение рабочего механизма; $R_{\text{я}}$ — сопротивление якоря двигателя; ψ — угол поворота вала двигателя; M_{δ} — момент, развиваемый двигателем; M_c — момент нагрузки; L_1 , R_1 , I_1 , E_1 — индуктивность обмотки возбуждения генератора, его омическое сопротивление, ток и электродвижущая сила (ЭДС) соответственно; L_2 , R_2 , I_2 , E_2 — аналогичные величины обмотки возбуждения двигателя; E_2 — ЭДС в контуре якоря двигателя, связанная с током в его обмотке возбуждения

выражением $E_2 = \Phi_2(I_1)$; $E_0 = c I_2 \dot{\psi}$ — ЭДС в контуре якоря двигателя, где $c = const$ — коэффициент пропорциональности.

Во-первых, при заданных начальных условиях требуется найти закон изменения во времени ЭДС $E_1(\tau) = E_1^*(\tau)$, который минимизирует значение функционала

$$F_1(E_1) = \int_0^{\hat{\tau}} (I_{\text{я}}^2(E_1, \tau) R_{\text{я}} + I_1^2(E_1, \tau) R_1 + I_2^2(E_1, \tau) R_2) d\tau, \quad (8)$$

имеющего смысл суммарных затрат энергии на перемещение якоря двигателя за период времени $[0; \hat{\tau}]$.

Во-вторых, функция $E_1^*(\tau)$ должна обеспечивать поворот якоря двигателя на требуемый угол ψ_{max} за указанный период времени. Другими словами, эта функция должна обеспечивать минимальное значение функционала

$$F_2(E_1) = abs(\psi(E_1(\hat{\tau})) - \psi_{\text{max}}). \quad (9)$$

В-третьих, функция $E_1^*(\tau)$ должна обеспечивать нулевую угловую скорости якоря двигателя в момент времени $\hat{\tau}$, то есть минимизировать функционал

$$F_3(E_1) = abs(\dot{\psi}(E_1(\hat{\tau}))). \quad (10)$$

Таким образом, имеем трехкритериальную задачу оптимального управления: найти допустимое управление $E_1^*(\tau) \in D_E$, которое на решениях системы уравнений (6), (7) обеспечивает минимум критериев оптимальности (8) - (10).

При всех известных недостатках решения задачи оптимального управления методом сведения ее к задаче нелинейного программирования, используем именно этот метод [10]. Обозначим для простоты записи $E_1(\tau) = u(\tau)$ и положим, что множество допустимых управлений $D_E = D_u$ определяют двухсторонние ограничения вида $u^- \leq u(\tau) \leq u^+$. Покроем интервал $[0; \hat{\tau}]$ равномерной сеткой, узлы которой обозначим τ_i , $i \in [0; |U|]$.

Будем искать оптимальное управление $u^*(\tau)$ в классе кусочно-линейных функций (рисунок 16).

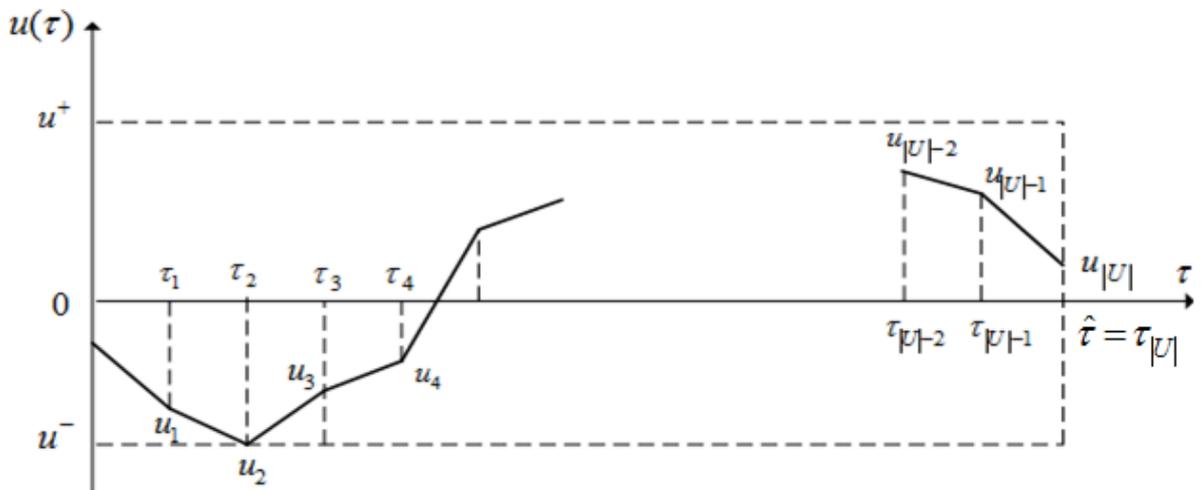


Рисунок 16 - Кусочно-линейная аппроксимация управления $E_1(\tau) = u(\tau)$

Обозначив $U = (u_1, u_2, \dots, u_{|U|})$ - $(|U| \times 1)$ -вектор, где $u_i = u(\tau_i)$, получим трехкритериальную задачу вида

$$\min_{U \in D_U} F_j(U) = F_j(U^*), \quad j = 1, 2, 3; \quad (11)$$

$$D_U = \{u^- \leq u_i \leq u^+, i \in [1:|U|]\}.$$

Переведем критерии оптимальности $F_1(U)$, $F_2(U)$ в ограничения с помощью алгоритма штрафных функций, то есть перейдем от задачи (11) к одноцелевой задаче глобальной безусловной оптимизации

$$\min_{U \in R^{|U|}} \varphi(U) = \min_{U \in D_U} (F_1(U) + \alpha_1 F_2(U) + \alpha_2 F_3(U)), \quad (12)$$

где α_1, α_2 - коэффициенты штрафа.

Примем следующие значения параметров задачи: $J=0,425$ кг·м²; $R_\gamma=0,224$ Ом; $M_\delta = k_\delta I_2 I_\gamma$ Нм, где $k_\delta=10$ - масштабный коэффициент; $M_c = 150$ Н·м; $L_1 = 8$ Гн; $R_1 = 43$ Ом; $L_2 = 8$ Гн; $R_2 = 43$ Ом; $E_2 = 40$ В; $\Phi_2(I_1) = k_2 I_1$, где $k_2=10$ - коэффициент кривой намагничивания; $c=1$;

$u^- = 0$; $u^+ = 100$ В. Начальные условия примем нулевыми: $\psi^0 = \dot{\psi}^0 = 0$; $I_1^0 = I_2^0 = 0$. Конечное состояние угла поворота якоря двигателя полагаем равным $\psi_{\max} = 210, \text{град}$. Движение системы рассматриваем на интервале $[0; 20]$, то есть полагаем $\hat{t} = 20$ с.

Число узлов сетки, покрывающей интервал $[0; \hat{t}]$, принимаем равным 21, так что размерность вектора варьируемых параметров $|U| = 20$.

Потребуем удовлетворения ограничений

$$F_1(E_1) = |\psi(E_1, \hat{t}) - \psi_{\max}| = 0, \quad (13)$$

$$F_2(E_1) = |\dot{\psi}(E_1, \hat{t})| = 0 \quad (14)$$

с точностями $\varepsilon_1 = 1,0$, $\varepsilon_2 = 1,0$ соответственно. Начальные значения коэффициентов штрафа α_1, α_2 принимаем равными 2,0; 5,0.

4.1. Алгоритм CIAC-S

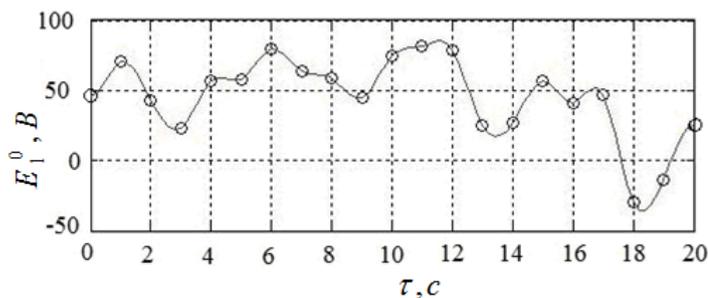
Принимаем следующие значения основных свободных параметров алгоритма: размер популяции равен $|S| = 300$; максимальное число поколений $\hat{t} = 100$; коэффициент устойчивости феромонных точек $\rho = 0,1$; минимальное количество феромона в точке $\theta_{\min} = 0,0001$; максимальное число сообщений $\mu_{\max} = 30$; делитель размерного параметра для всех агентов одинаков и равен

$$\psi = \text{abs}(N_1(0; \sigma)),$$

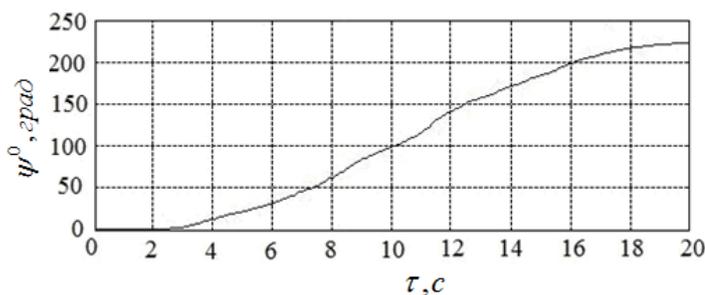
где $N_1(0; \sigma)$ - случайная величина, распределенная по нормальному закону с нулевым математическим ожиданием и средним квадратичным отклонением $\sigma = 0,9$.

Пример начального приближения к решению иллюстрирует рисунок 17. Указанному приближению соответствуют угол $\psi(\hat{t}) = 223,14^\circ$, угловая скорость $\dot{\psi}(\hat{t}) = 2,46 \frac{\text{град}}{\text{с}}$ и расход энергии $F_1(E_1^0) = 5636 \text{ Вт}$.

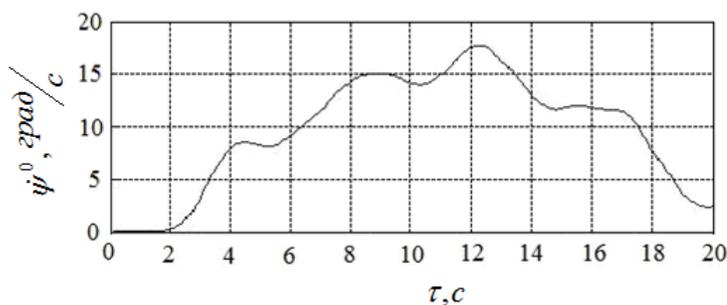
Лучшие (с точки зрения достигнутого значения функционала (8)) результаты вычислительного эксперимента иллюстрируют рисунки 18а – 18в. Рисунок 18а показывает найденное оптимальное управление $E_1^*(\tau)$, а рисунки 18б, 18в – соответствующие законы изменения во времени угла поворота якоря двигателя $\psi^*(\tau)$ и его скорости $\dot{\psi}^*(\tau)$.



а) $E_1^0(\tau)$

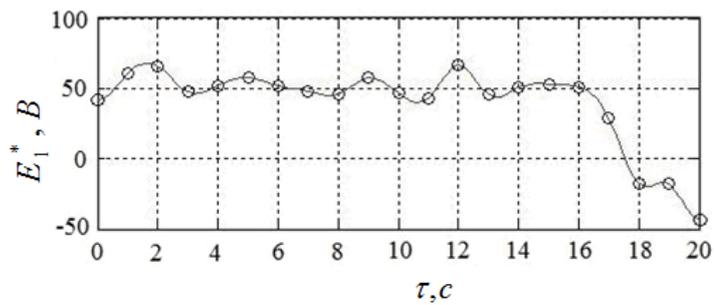


б) $\psi^0(\tau)$

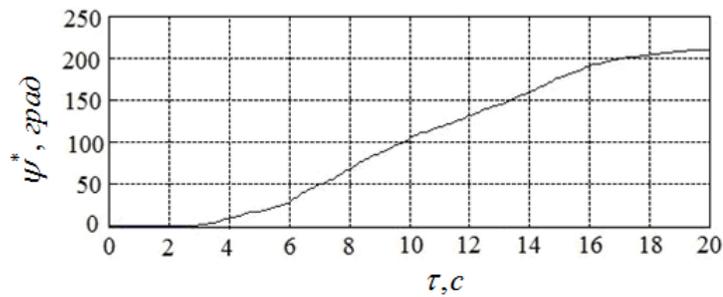


в) $\dot{\psi}^0(\tau)$

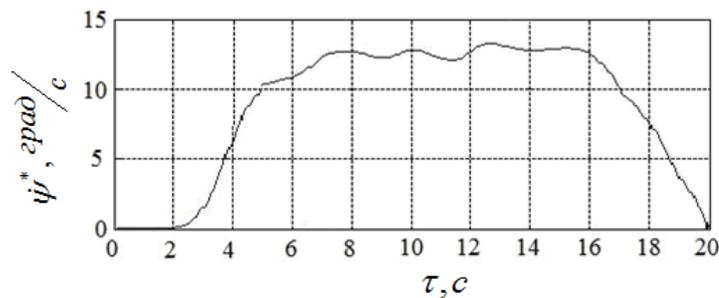
Рисунок 17 – Пример начального приближения к решению $E_1^0(\tau)$ и соответствующие функции $\psi^0(\tau)$, $\dot{\psi}^0(\tau)$; $|U| = 21$



а) $e_1^*(\tau)$



б) $\varphi^*(\tau)$



в) $\dot{\varphi}^*(\tau)$

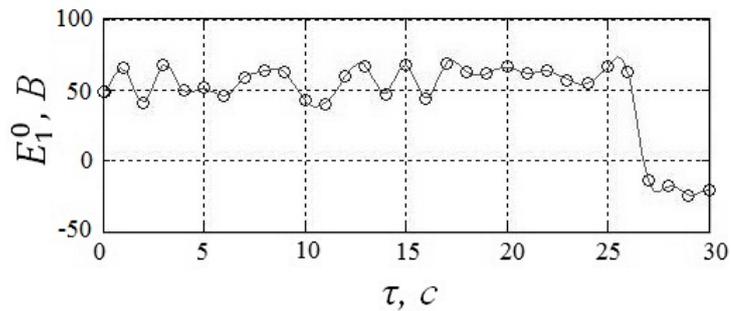
Рисунок 18 – Результаты решения задачи – оптимальная функция $E_1^*(\tau)$ и соответствующие функции $\psi^*(\tau)$, $\dot{\psi}^*(\tau)$; $|U| = 21$; алгоритм *CIAC-S*

Полученному решению задачи соответствуют угол $\psi^*(\hat{\tau}) = 209,7^\circ$, угловая скорость $\dot{\psi}^*(\hat{\tau}) = 0,02 \text{ град}/\text{с}$ и расход энергии $F_1(E_1^*) = 3352 \text{ Вт}$. Таким образом, требование (13) выполнено с точностью $0,3^\circ$, требование (14) – с точностью $0,02 \text{ град}/\text{с}$, а расход энергии уменьшился по сравнению с начальным приближением более чем на 40 %.

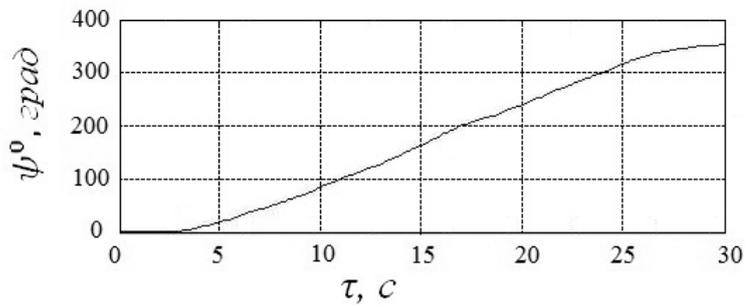
Рассмотрим также движение системы на интервале $[0; 30]$ секунд, когда конечный угол поворота якоря двигателя равен $\psi_{\max} = 330^\circ$ м. Число узлов

сетки, покрывающей интервал $[0; \hat{\tau}]$, принимаем равным 31, так что размерность вектора варьируемых параметров $|U| = 31$.

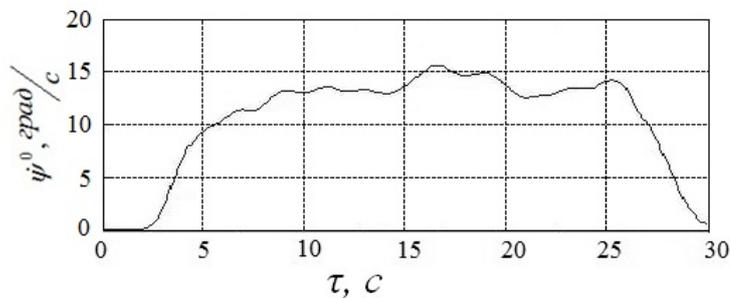
Пример начального приближения к решению в этом случае иллюстрирует рисунок 19. Указанному приближению соответствуют угол $\psi(\hat{\tau}) = 352,45^\circ$, угловая скорость $\dot{\psi}(\hat{\tau}) = 0,75 \text{ град}/\text{с}$ и расход энергии $F_1(E_1^0) = 6574 \text{ Вт}$.



а) $E_1^0(\tau)$



б) $\psi^0(\tau)$



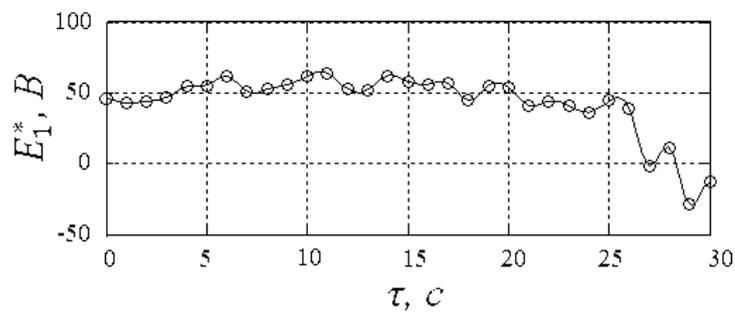
в) $\dot{\psi}^0(\tau)$

Рисунок 19 – Пример начального приближения к решению $E_1^0(\tau)$ и соответствующие функции $\psi^0(\tau)$, $\dot{\psi}^0(\tau)$; $|U| = 31$

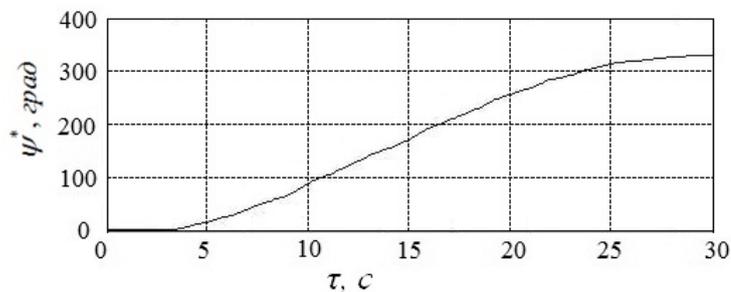
Лучшие результаты вычислительных экспериментов, выполненных при помощи алгоритма *CIAC-S*, представлены на рисунке 20. Ему соответствует угол $\psi^*(\hat{\tau}) = 330,15^\circ$, угловая скорость вала $\dot{\psi}^*(\hat{\tau}) = 0,12 \text{ град}/\text{с}$ и расход энергии $F_1(E_1^*) = 4083 \text{ Вт}$.

4.2. Алгоритм *НСIAC-M*

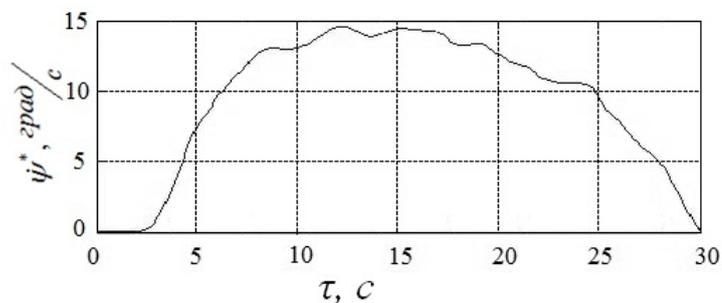
Максимальное число оценок целевой функции в процессе локального поиска ограничиваем величиной $n_f^{\max} = 400$, число запусков алгоритма локального поиска полагаем равным $h = 200$.



а) $E_1^0(\tau)$



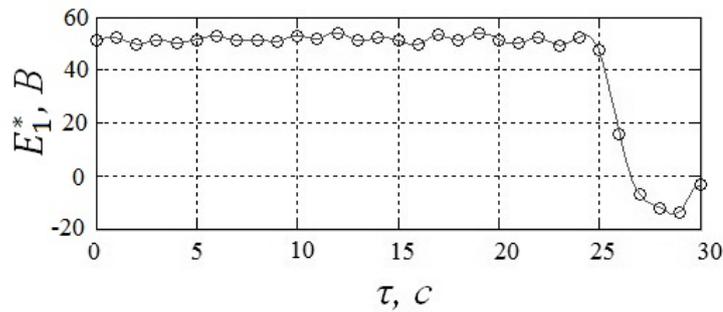
б) $\psi^0(\tau)$



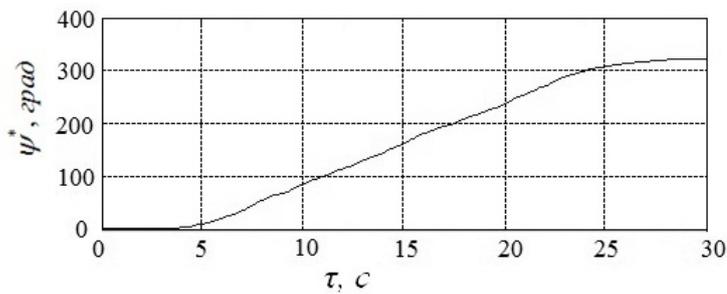
в) $\dot{\psi}^0(\tau)$

Рисунок 20 – Результаты решения задачи – оптимальная функция $E_1^*(\tau)$ и соответствующие функции $\psi^*(\tau)$, $\dot{\psi}^*(\tau)$; $|U| = 31$; алгоритм *CIAC-S*

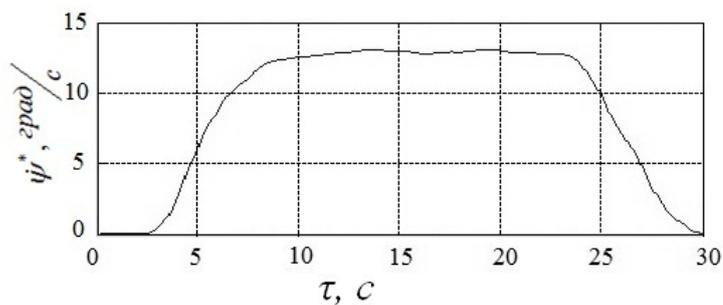
Результаты решения задачи с помощью алгоритма *НСИАС-М*, полученные в условиях п.4.1, иллюстрирует рисунок 21. Представленному решению соответствует угол $\psi^*(\hat{\tau}) = 329,98^\circ$, угловая скорость вала $\dot{\psi}^*(\hat{\tau}) = 7,24 \cdot 10^{-2} \text{ град}/\text{с}$ и расход энергии $F_1(E_1^*) = 3895 \text{ Вт}$.



а) $E_1^0(\tau)$



б) $\psi^0(\tau)$



в) $\dot{\psi}^0(\tau)$

Рисунок 21 – Результаты решения задачи – оптимальная функция $E_1^*(\tau)$ и соответствующие функции $\psi^*(\tau)$, $\dot{\psi}^*(\tau)$; $|U| = 31$; алгоритм *НСИАС-М*

Из рисунков 20, 21 следует, что, как и ожидалось, алгоритм *НСИАС-М* обеспечивает повышение точности решения и позволяет получить более

гладкий управляющий сигнал. Вследствие увеличения числа испытаний скорость поиска замедляется по сравнению с алгоритмом *CIAC-S* примерно на 30%.

Заключение

Представленные результаты исследования позволяют сделать вывод, что предложенная мультимемеевая гибридизация муравьиного алгоритма *NCIAC* позволяет не только увеличить вероятность локализации глобального экстремума целевой функции, но и повысить точность решения. Кроме того, за счет повышения эффективности поиска, гибридный алгоритм дает возможность уменьшить число агентов в системе и повысить тем самым его быстродействие. Мультимемеевая гибридизация приводит к алгоритму, обладающему свойствами самоадаптации, поскольку в процессе поиска меммы конкурируют между собой и победивший на данном этапе поиска мем используется на последующих этапах.

В развитие работы авторы планируют использовать расширенный набор меммов, выбирать наилучших меммов для различных подобластей пространства поиска. Планируется также параллельная реализация рассматриваемых алгоритмов.

Список литературы

1. Карпенко А.П., Чернобривченко К.А. Эффективность оптимизации методом непрерывно взаимодействующей колонии муравьев (*CIAC*) // Наука и образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2011. № 2. Режим доступа: <http://technomag.edu.ru/doc/165551.html> (дата обращения 20.10.2012).
2. Dréo J., Siarry P. Continuous interacting ant colony algorithm based on dense heterachy // Future Generation Computer Systems. 2004. Vol. 20, no. 5. P. 841–856. DOI: <http://dx.doi.org/10.1016/j.future.2003.07.015>

3. Dreco J., Siarry P. Hybrid Continuous Interacting Ant Colony aimed at enhanced global optimization // *Algorithmic Operations Research*. 2007. Vol. 2, no. 1. P. 52–64.
4. Krasnogor N. Studies on the Theory and Design Space of Memetic Algorithms : Ph.D. Thesis. Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, U.K, 2002. 289 p.
5. Karpenko A.P., Svianadze Z.O. Meta-optimization based on self-organizing-map and genetic algorithm // *Optical Memory and Neural Networks (Information Optics)*. 2011. Vol. 20, No. 4. P. 279-283. DOI: 10.3103/S1060992X11040059
6. Tang K., Yao X., Suganthan P.N., MacNish C., Chen Y.P., Chen C.M., Yang Z. Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization. Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.
7. Александров А.Г. Оптимальные и адаптивные системы. М.: Высшая школа, 1989. 263 с.
8. Hu X.-M., Zhang J., Li Y. Orthogonal Methods Based Ant Colony Search for Solving Continuous Optimization Problems // *Journal of Computer Science and Technology*. 2008. Vol. 23, no. 1. P. 2-18. DOI: 10.1007/s11390-008-9111-5
9. MatLab, Optimization Toolbox Fminunc. Центр компетенции MathWorks. Режим доступа: <http://matlab.exponenta.ru> (дата обращения 20.10.20120).
10. Федоренко Р.П. Приближенное решение задач оптимального управления. М.: Наука, 1978. 488 с.

Multimemeev modified hybrid ant algorithm for HCIAC continuous optimization

09, September 2012

DOI: 10.7463/0912.0470529

Karpenko A.,P., Chernobrivchenko K., A.

Russia, Bauman Moscow State Technical University

apkarpenko@mail.ruchernobrivchenko.k@gmail.com

This publication continues the authors' work in the field of ant algorithms of global continuous optimization. The authors propose a modification of the known HCIAC algorithm which was proposed by J.Dreo and P. Syarri in 2007. This modification is based on a set of memes in order to make local search more efficient. The authors present the results of their research on effectiveness of the modified algorithm - HCIAC-M – during the search of global extremum of Rosenbrock and Rastrigin functions. As an example, the authors consider practical importance of optimization of the engine-generator system.

Publications with keywords: [global optimization](#), [ant algorithm](#), [mem](#)Publications with words: [global optimization](#), [ant algorithm](#), [mem](#)

References

1. Karpenko A.P., Chernobrivchenko K.A. Effektivnost' optimizatsii metodom nepreryvno vzaimodeistvuiushchei kolonii murav'ev (CIAC) [Efficiency of optimization by continuous interacting ant colony (CIAC)]. *Nauka i obrazovanie MGTU im. N.E. Baumana* [Science and Education of the Bauman MSTU], 2011, no. 2. Available at: <http://technomag.edu.ru/doc/165551.html> , accessed 20.10.2012.
2. Dréo J., Siarry P. Continuous interacting ant colony algorithm based on dense heterachy. *Future Generation Computer Systems*, 2004, vol. 20, no. 5, pp. 841-856. DOI: <http://dx.doi.org/10.1016/j.future.2003.07.015>
3. Dreo J., Siarry P. Hybrid Continuous Interacting Ant Colony aimed at enhanced global optimization. *Algorithmic Operations Research*, 2007, vol. 2, no. 1, pp. 52-64.
4. Krasnogor N. *Studies on the Theory and Design Space of Memetic Algorithms. Ph.D. Thesis*. Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, U.K, 2002. 289 p.

<http://technomag.edu.ru/doc/470529.html>

5. Karpenko A.P., Svianadze Z.O. Meta-optimization based on self-organizing-map and genetic algorithm. *Optical Memory and Neural Networks (Information Optics)*, 2011, vol. 20, no. 4, pp. 279-283. DOI: 10.3103/S1060992X11040059
6. Tang K., Yao X., Suganthan P.N., MacNish C., Chen Y.P., Chen C.M., Yang Z. *Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization*. Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.
7. Aleksandrov A.G. *Optimal'nye i adaptivnye sistemy* [Optimal and adaptive systems]. Moscow, Vysshiaia shkola, 1989. 263 p.
8. Hu X.-M., Zhang J., Li Y. Orthogonal Methods Based Ant Colony Search for Solving Continuous Optimization Problems. *Journal of Computer Science and Technology*, 2008, vol. 23, no. 1, pp. 2-18. DOI: 10.1007/s11390-008-9111-5
9. *MatLab, Optimization Toolbox Fminunc. Tsentr kompetentsii MathWorks* [MatLab, Optimization Toolbox Fminunc. Competence center MathWorks]. Available at: <http://matlab.exponenta.ru> , accessed 20.10.2012.
10. Fedorenko R.P. *Priblizhennoe reshenie zadach optimal'nogo upravleniia* [Approximate solution of optimal control problems]. Moscow, Nauka, 1978. 488 p.