УДК 004.023

ИССЛЕДОВАНИЕ АЛГОРИТМА СЕРЫХ ВОЛКОВ

В.А. ЧАСТИКОВА, С.А. ЖЕРЛИЦЫН

Кубанский государственный технологический университет,
350072, Российская Федерация, г. Краснодар, ул. Московская, 2;
электронная почта: kpytooooo@gmail.com

В данной статье проведен анализ и исследование одного из алгоритмов роевого интеллекта - алгоритма серых волков. Сформулировано его детальное описание, рассмотрен способ модификации алгоритма для повышения эффективности его работы на примере задачи глобальной оптимизации. Для исследования алгоритм был реализован в классическом виде с задаваемым пользователем количеством волков, измерений пространства, числом итераций и промежутком значений, на котором ведется поиск. Глобальная оптимизация проводилась на примере функций Де Джонга, Растригина и Розенброка. По итогам проведенных исследований можно сделать вывод, что алгоритм серых волков эффективен для поиска глобального оптимума как функции с одним экстремумом, так и много экстремальных функций, а его гибкость дает возможность практического применения алгоритма максимально эффективно для каждого конкретного случая.

Ключевые слова: алгоритм серых волков, эвристический алгоритм, роевой интеллект, глобальная оптимизация.

При построении сложной системы необходимо рационально использовать имеющиеся ресурсы, а значит, оптимизировать ее работу. Однако иногда становится невозможно применить классические методы для решения конкретной задачи. Причинами этому может стать сложность вычислений, их большое количество, существенная нелинейность или невозможность описать алгоритм работы. Именно для таких ситуаций была разработана модель метаэвристики. Этот термин обозначает алгоритм решения задачи, не имеющий строгого обоснования, но, тем не менее, способный выдать за адекватное время приемлемые, близкие к оптимальным, результаты в большем количестве значимых случаев[1, 4].

Алгоритм серых волков – метаэвристический алгоритм роевого интеллекта, ориентированный на оптимизацию функции в векторном пространстве. В его основе лежит механизм охоты серых волков в природе. Все агенты (боиды) делятся на 4 иерархических типа: альфа, бета, дельта и омега.

Для выполнения оптимизации выполняются 3 основных этапа, в ходе которых волки ищут добычу, окружают, а затем атакуют её [3,5].

Математическая реализация социальной иерархии заключается в том, что тройке самых приспособленных, то есть близких к искомой точке, экземпляров присваиваются ранги альфа, бета идельта, по мере убывания приспособленности. Остальные волки получают ранг омега.

В основном цикле алгоритма волки окружают свою добычу. Модель окружения добычи описывается следующими формулами:

$$\bar{D} = |\bar{C} * \bar{X}_p(t) - \bar{X}(t)|$$

$$\bar{X}(t+1) = \bar{X}_p(t) - \bar{A} * \bar{D}$$

Где t – текущая итерация, A и C – вспомогательные коэффициенты векторов, X_p – вектор положения одного из 3-х лучших волков, X – положение рассматриваемого волка.

Векторы A и C рассчитываются следующим образом:

$$\bar{A} = 2\bar{a} * \bar{r}_1 - \bar{a}$$

$$\bar{C} = 2 * \bar{r}_2$$

Причём a – линейно уменьшается с 2 до 0 в течение всех итераций, а r_1и r_2 – случайные числа из отрезка [0;1].

Таким образом, агенты распознают местоположение жертвы по ее окружению.

$$\bar{D}_\alpha = |\bar{C}_1 * \bar{X}_\alpha - \bar{X}|$$

$$\bar{D}_\beta = |\bar{C}_2 * \bar{X}_\beta - \bar{X}|$$

$$\bar{D}_\delta = |\bar{C}_3 * \bar{X}_\delta - \bar{X}|$$

$$\bar{X}_1 = \bar{X}_\alpha - \bar{A}_1 * (\overline{D_\alpha})$$

$$\bar{X}_2 = \bar{X}_\beta - \bar{A}_2 * (\overline{D_\beta})$$

$$\bar{X}_3 = \bar{X}_\delta - \bar{A}_3 * (\overline{D_\delta})$$

$$\bar{X}(t+1) = \frac{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}{3}$$

С помощью этих уравнений каждый волк обновляет свою позицию в соответствии сальфа, бета и дельта волками, попадая в случайное место в пределах круга, определяемого вышеперечисленными волками и приближаясь к добыче (рис.1). [2]

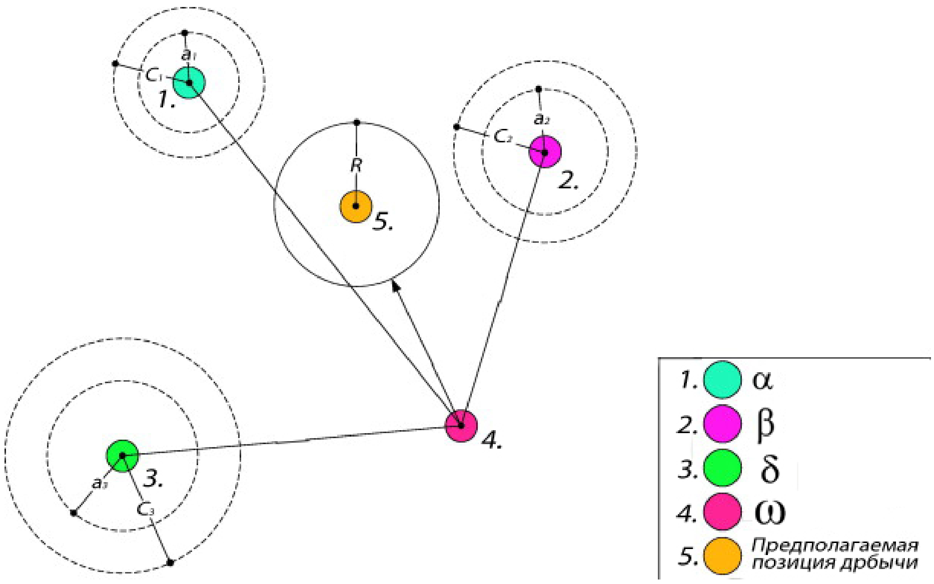

Рисунок 1 – Обновление позиции волка

Так как коэффициент *а*уменьшается с каждой итерацией и от него зависит вектор *A*, поиск оптимума заданной функции делится на две фазы: разведка и атака. Процесс поиска отделяется от процесса атаки тем, что в первом случае модуль вектора*A*больше единицы, и волки, как правило, отклоняются от искомой точки, а во втором – меньше, и волки сходятся к добыче.

В общем виде реализуется следующий алгоритм[2]:

1) Инициализировать позиции серых волков популяции X_i (i= 1, 2, ..., n)

2) Инициализировать a, A и C

3) Вычислить пригодность каждого поискового агента

4)X_α = Лучший поисковый агент

5)X_β = Второй лучший поисковый агент

6)X_δ = Третий лучший поисковый агент

7) Пока (t <Максимальное количество итераций)

8) Для каждого поискового агента: обновление позиции текущего поискового агента по приведенным выше уравнениям

9) Обновить значения a, A и C

10) Вычислить пригодность каждого из волков

11) Обновить X_α, X_β, а также X_δ

12) $t = t + 1$

13) Конец «Пока»

14) Вывод X_α

В ходе исследований данный алгоритм был реализован в классическом виде для решения задачи глобальной оптимизации некоторых функций. Перед запуском поиска оптимума реализована возможность задать параметры работы алгоритма, а именно: количество измерений, в которых задана функция, число волков, выполняющих поиск, и количество итераций. Это дает возможность оценить эффективность работы алгоритма серых волков на примере выбранных функций.

1. Отклонение от минимума при оптимизации функции Де Джонга в трёхмерном пространстве (по горизонтали – количество итераций, по вертикали – число поисковых агентов):

	25	50	75	100	150	200	250	500	1000
	27,3	5,07	0,211	0,035	10^{-4}	10^{-7}	10^{-13}	10^{-30}	10^{-46}
	2,53	10^{-5}	10^{-10}	10^{-11}	10^{-17}	10^{-23}	10^{-28}	10^{-55}	10^{-118}
0	10^{-3}	10^{-8}	10^{-12}	10^{-17}	10^{-20}	10^{-27}	10^{-30}	10^{-72}	10^{-142}
5	10^{-5}	10^{-14}	10^{-21}	10^{-30}	10^{-38}	10^{-55}	10^{-68}	10^{-125}	10^{-254}

2. Отклонение от минимума при оптимизации функции Растригина в трёхмерном пространстве (по горизонтали – количество итераций, по вертикали – число поисковых агентов):

	25	50	75	100	150	200	250	500	1000
	41,63	0,19	0,04	0,007	10^{-5}	10^{-7}	10^{-11}	0	0
	13,48	9,3	9,3	0,5	10^{-9}	10^{-12}	0	0	0
0	3,03	3,03	2,13	1,23	10^{-6}	10^{-13}	0	0	0
5	10^{-4}	10^{-10}	10^{-13}	0	0	0	0	0	0

3. Отклонение от минимума при оптимизации функции Розенброка в трёхмерном пространстве (по горизонтали – количество итераций, по вертикали – число поисковых агентов):

	25	50	75	100	150	200	250	500	1000
	12,8	12,8	12,8	9,51	9,51	8,23	8,23	0,99	0,99
	181,6	36,68	17,3	17,3	17,3	17,3	17,3	17,3	17,3
0	3,28	2,11	0,41	0,41	0,41	0,41	0,41	0,41	0,41
5	11	11	11	9,89	9,89	0,99	0,99	0,99	0,99

По результатам проведенных исследований можно сделать вывод, что алгоритм серых волков отлично справляется с поиском минимума функций как с одним экстремумом, так и с множеством, однако в процессе оптимизации функции, плавно убывающей в некоторой области, но стремительно уменьшающейся в любой другой точке пространства, алгоритм имеет довольно высокую погрешность вычислений.

Как говорилось выше, поиск оптимума делится на две фазы, балансировку которых можно настраивать, модифицируя алгоритм. В исходном виде коэффициент *a*вычислялся по формуле:

$$a = 2 * \left(1 - \frac{t}{T}\right),$$

то есть линейно убывал с каждой новой итерацией. При следующем задании формулы:

$$a = 2 * \left(1 - \frac{t^2}{T^2}\right)$$

данный коэффициент будет больше единицы в течение 70% времени работы алгоритма, то есть на разведку, поиск искомого оптимума уйдет большая часть ресурсов, а точность результата уменьшится, так как увеличив время поиска оптимума, урезается количество итераций атаки до 30% от отведенного максимума.

Руководствуясь тем же принципом, можно увеличить длительность атаки в ущерб разведке для повышения точности найденного оптимума, но уменьшив при этом вероятность того, что оптимум глобальный.

ЛИТЕРАТУРА

1. Luke S. Essentials of Metaheuristics / Luke S. – Second edition, Online version 2.1, 2014. – сс. 9-12.

2.Algorithm models / Grey Wolf Optimizer – [Электронныйресурс] – 2016 - https://en.wikiversity.org/wiki/Algorithm_models/Grey_Wolf_Optimizer

3. S. Mirjalili, A. Lewis. Grey Wolf Optimizer.Advanced in Engineering Software / S. Mirjalili, A. Lewis. - 2014. сс. 46-61.

4. Частикова В.А., Власов К.А.Разработка и сравнительный анализ эвристических алгоритмов для поиска наименьшего гамильтонова цикла в полном графе// Фундаментальные исследования. 2013. № 10-1. С. 63-67.

5. В. Баранюк, О. Смирнова. Роевой интеллект как одна из частей онтологической модели бионических технологий/ В. Баранюк, О. Смирнова. - InternationalJournalofOpenInformationTechnologiesISSN: 2307-8162 vol. 3, no. 12, 2015.

REFERENCES

1. Luke S. Essentials of Metaheuristics / Luke S. – Second edition, Online version 2.1, 2014. – ss. 9-12.

2. Algorithm models / Grey Wolf Optimizer – [Elektronnyyresurs] – 2016 - https://en.wikiversity.org/wiki/Algorithm_models/Grey_Wolf_Optimizer

3. S. Mirjalili, A. Lewis. Grey Wolf Optimizer. Advanced in Engineering Software / S. Mirjalili, A. Lewis. - 2014. ss. 46-61.

4.Chastikova V.A., Vlasov K.A. Razrabotka i sravnitelnyyanaliz evristicheskikh algoritmov dlya poiska naimenshegogamiltonova tsikla v polnom grafe. Fundamentalnye issledovaniya,no. 10, 2013. pp. 63–67.

5. V. Baranyuk, O. Smirnova. Roevoy intellect kak odna iz chastey ontologicheskoy modeli bionicheskikh tekhnologiy / V. Baranyuk, O. Smirnova. - International Journal of Open Information Technologies ISSN: 2307-8162 vol. 3, no. 12, 2015.

RESEARCHOF THE GREY WOLF ALGORITHM

V.A. CHASTIKOVA, S.A. ZHERLITSIN

Kuban State Technological University,
2, Moskovskaya st., Krasnodar, Russian Federation, 350072;
E-mail: kpytooooo@gmail.com

This article presents the results of analysis and research of Grey Wolf algorithm as well as its detailed description. A method of modifying the algorithm has been examined in the case of global optimization problem in order to improve its efficiency. The algorithm has been implemented in the classical form with customer-specified number of wolves, space dimensions, number of iterations and interval of values on which the search is conducted. Global optimization has been tested in the case of De Jong, Rastrigin and Rosenbrock functions. According to the results of the research, the Grey Wolf algorithm proves its efficiency for finding a global optimum for both single-extremum and multi-extremum functions. Its flexibility makes the algorithm able to be adjusted to any specific case thus enabling it to work most effectively.

Key words: gray wolves algorithm, heuristic algorithm, swarm intelligence, global optimization.