Gradient-based Ant Colony Optimization for Continuous Spaces

M. Eftekhari, B. Daei, and S. D. Katebi
Department of Computer Science and Engineering, School of Engineering, Shiraz University

Abstract: A novel version of Ant Colony Optimization (ACO) algorithms for solving continuous space problems is presented in this paper. The basic structure and concepts of the originally reported ACO are preserved and adaptation of the algorithm to the case of continuous space is implemented within the general framework. The stigmergic communication is simulated through considering certain direction vectors which are memorized. These vectors are normalized gradient vectors that are calculated using the values of the evaluation function and the corresponding values of object variables.

The proposed Gradient-based Continuous Ant Colony Optimization (GCACO) method is applied to several benchmark problems.
کلمه های کلیدی: جفت انتخاب، الگوریتم، تصادفی، سیستم حمل و نقل، محدودیت.
سیستم مورچه را برطرف کرده و کارایی مطلوب را داشته باشد. از این مرحله به بعد هرگونه عبارت گروهی به همراه اجتماع مورچه ساخته شده است. منظور ساختار کلی الگوریتم سیستم اجتماع مورچه است. 

در زمینه تعمیم الگوریتم های سازی اجتماع مورچه با فضای پیوسته تحقیقات ایندیکس صورت گرفت است که اولین آنها توسط بیلبورف [16] در سال 1995 انجام شد. به همراهی که پیشنهاد شده است، به ترتیب در [14-16] اورده شده است. در اکثر این روش ها جستجو در دو مرحله اساسی و محل انجام می شود. که مرحله اساسی آن یک تری شده است. در این مقاله، الگوریتم پیشنهادی بر مبنای چهار گروه کلی الگوریتم سیستم اجتماع مورچه است که به فضای پیوسته تعمیم داده شده است.

۲- روش پیشنهادی، تعمیم الگوریتم پیشنهادی سازی اجتماع مورچه به فضای پیوسته

در این بخش ما به توضیح الگوریتم پیشنهادی می پردازیم. در شکل (۱) جریان الگوریتم و شبکه کد به همراه سازی اجتماع مورچه پیشنهادی که با اختصار GCACO نامیده می شود، آورده شده است. سپس موارد تعمیم و اصلاح الگوریتم برای فضای پیوسته شرح داده می شود.

اولین تفاوت این است که در الگوریتم پیشنهادی برای هر اجتماع یک شعاع حرکت $R_c$ در نظر گرفته شده است. (برای اولین اجتماع متقاعد شعاع یک در نظر گرفته می شود). سپس متقاعد شعاع توسط یک تابع نامی به شکل زیر کاهش پیدا می کند.

$$R_c = R_c e^{-\frac{t}{T_{\text{max}}}}$$  

(۴)

تعداد تکرارها $T_{\text{max}}$ تقلب الگوریتم GCACO، الگوریتم پیشنهادی اجتماع مورچه که به عنوان یک بعدی در نظر گرفته می شود.

کرد. حال در تکرار $t$ احتمال این که مورچه $k$ ام نقطه $x$ از $N_k$ نقطه $x$ ام را برای رفتگی انتخاب کند، در حقیقت $k$ ام نقطه $x$ ام است، به صورت زیر است. 

$$p_{k}^{(x)}(t) = \frac{a_{k}^{(x)}(t)}{\sum_{i=1}^{N_k} a_{i}^{(x)}(t)}$$  

(۲)

در معادله (۲) $p_{k}^{(x)}(t)$ است و $\sum_{i=1}^{N_k} a_{i}^{(x)}(t)$ نقاطه است که توسط مورچه $k$ ام هر زمان شرایطی مشابه دارد. این تفکیک با استفاده از فهرست منونی قابل تکرار است.

ضای شدن متقاعد فرمون و همچنین تبخیر فرمون با معادله زیر در هر تکرار صورت می‌گیرد.

$$\Delta r_{ij}^{(x)} = \sum_{k=1}^{m} \Delta r_{ij}^{(x)} \pm (\Delta t_{ij}^{(x)} - \Delta t_{ij}^{(x)})$$

(۳)

در معادله (۳) $\Delta r_{ij}^{(x)}$ متقاعد فرمون $ij$ است که توسط مورچه $k$ ام روی بال (یا $ij$) نشاست کرده است و $\Delta t_{ij}^{(x)}$ متقاعد لكل فرمون $ij$ است.

نتیجه روش بالا (یا الگوریتم گروهی مورد شناخته شده) گروه به سمت انتقال از طرفی خیلی جالب و از طرف دیگر خیلی نامید کنندگی است. الگوریتم سیستم مورچه قادر به یک نتیجه مثبت می‌باشد که با انتقال دو کره به دو نقطه توسط الگوریتم پیشنهادی با تفاوت اولیه را به‌طور یکسان پیدا می‌نماید. این تفاوت اساسی دارد که در [17] به تفصیل داده می‌شود. 

این الگوریتم، بهترین الگوریتم پیشنهادی اجتماع مورچه تاکنون است، که توانسته است تقاضای مربوط به الگوریتم‌های

35

استقلال سال ۲۵، شماره ۱، شهروند ۱۳۸۵
ان خاصیت گروه‌ای آن را به روش‌های حریصانه جستجو مانند ته نوردی تزدیک ساخته است.
تیک خوردن که در خط ۶ از شکل (۲) آمده است، توسط معادله زیر انجام می‌شود:
\[
\text{new} = \text{evap\_factor} \times \text{old}
\]
عمل تیک خوردن زمانی اتفاق می‌افتد که تابع ارزیابی تغییری پیدا نمی کند. (این به معنی آن است که الگوریتم احتمالاً دچار رکود شده است و فرآیند سپاس به فرآیند شده‌اند.)
با توجه به شکل (۲)، هر مورچه توسط تابع

شکل ۱- نمودار جریانی مربوط به الگوریتم پیشنهادی

(new_active_ant) به راه می‌افتد و یک جواب بر می‌گردد که بهترین جواب است که بی‌بیا کرده است. سپس از بین این جوابها، بهترین جواب سریاسی انتخاب می‌شود و به عنوان تیک گرفته شود. هر مورچه توسط مقدار متغیرهای فرآیند حافظه و new_active_ant) تابع (۵) جدول مربوط به تغییر می‌دهد. این مقادیر برای حرکت مورچه بی‌بیا استفاده می‌شود. فرآیند حافظه الگوریتم با الگوریتم بهبود سازی اجتماع مورچه‌ها این است که در اینجا حافظه بین همه مورچه‌ها مانند فرآیند و جدول مبادلات مشترک است.
1 Procedure GCACO_Meta_heuristic() 
2 Initialize_the_parameters_of_GCACO(); 
3 While ( t < T_{max} ) 
4 \quad [\text{Best_Eval Function, Best_solution}] = \text{ants_generation_and_activity}(); 
5 if \text{No_change_in_Eval_Function} 
6 \quad Evaporate_Pheromone(); 
7 endif 
8 select_global_solution_among_the_Best_ones(); 
9 Reduce_the_Radius_of_colony_Movement(); 
10 Set_the_best_solution_of_this_colony_as_the_Nest_of_Next_colony; 
11 t=t+1; 
12 endwhile 
13 endprocedure 

14 Procedure ants_generation_and_activity() 
15 Initialize_the_Radius_of_colony_by_Rc; 
16 While Ant_count < Total_Ants 
17 \quad [\text{best_fitness, best_location, routing_table, memory, } \tau] = \text{new_active_ant}(); 
18 select_global_location_among_the_best_ones; 
19 set_returned_\tau_memory_routing-table_for_the_next_ant_movement(); 
20 Ant_count= Ant_count+1; 
21 endwhile 
22 endProcedure 
23 procedure new_active_ant(); 
24 Initialize Ra = Rc, Nest = Nest_of_colony, \tau_0 = 2 , \varphi = 0.05 ; 
25 While (Active_ant ) 
26 do 
27 \quad q = \text{Uniform_rand}(0,1); 
28 if \ q \leq q_0 
29 \quad x_{new} = x_{old} + R_a \times U(-1,1) 
30 else 
31 \quad \text{best_index} = \text{compute_best_direction}(); 
32 \quad x_{new} = x_{old} + R_a \times \text{memory(best_index)} 
33 endif 
34 while( not_feasible ) 
35 if ( memory_is_used ) 
36 \tau_{new_{best_index}} = (1-\varphi) \times \tau_{old_{best_index}} + \varphi \times \tau_0 
37 if ( \eta_{best_index} < \eta_{current} \text{ and } f_{current} < f_{previous} ) 
38 \eta_{best_index} = | f_{current} - f_{previous} | 
39 endif 
40 else 
41 \eta_{current} = | f_{current} - f_{previous} | 
42 if ( \forall i \eta_i < \eta_{current} \text{ and } f_{current} < f_{previous} ) 
43 NGV = \text{calculate_the_gradient_and_normalize_it}; 
44 Bad_index = \text{compute_worst_direction}(); 
45 Memory(Bad_index) = NGV; 
46 \tau_{new_{Bad_index}} = (1-\varphi) \times \tau_{old_{Bad_index}} + \varphi \times \tau_0 ; 
47 \endprocedure 

در الگوریتم $\text{GCACO}$، تا زمانی که یک مورچه فعال است، حوسطد گره انتخاب یک قانون تصمیم‌گیری تصادفی که با الگوریتم بیشتر انتخاب می‌شود موردی برای می‌باشد. حوسطد این موردی از فضای جستجو کاملاً تصادفی با اساس بردارهای گرافیکی ذخیره شده در حافظه است. این عدد $q$ عدد هر گره انجام می‌شود. گراف مانندی که صورت می‌گیرد با حدود $1.1$ در می‌باشد. این عدد $q$ در این صورت حوسطد در فضای حافظه است. حافظه از تعداد محدودی جهت تکمیل شده است و در صورت لزوم با حدودی جهت انجام می‌باشد و حوسطد این موردی اساس این جهت انجام می‌شود و حوسطد به دست می‌آید. تغییر می‌دهد. در غیر این صورت حوسطد در فضای حافظه است. حافظه از تعداد محدودی جهت تکمیل شده است و در صورت لزوم با حدودی جهت انجام می‌باشد و حوسطد به دست می‌آید. تغییر می‌دهد. در غیر این صورت حوسطد در فضای حافظه است. حافظه از تعداد محدودی جهت تکمیل شده است و در صورت لزوم با حدودی جهت انجام می‌باشد و حوسطد به دست می‌آید. تغییر می‌دهد. در غیر این صورت حوسطد در فضای حافظه است. حافظه از تعداد محدودی جهت تکمیل شده است و در صورت لزوم با حدودی جهت انجام می‌باشد و حوسطد به دست می‌آید. تغییر می‌دهد. در غیر
جدول ۱- جزئیات توابع محک

<table>
<thead>
<tr>
<th>بهینه فرمول ریاضی</th>
<th>محصوله جستجو</th>
<th>بعد</th>
<th>نام تابع</th>
<th>شناسه</th>
</tr>
</thead>
<tbody>
<tr>
<td>۰</td>
<td>[-۱۰۰,۱۰۰]</td>
<td>۲**</td>
<td>Schaffer F6</td>
<td>T1</td>
</tr>
<tr>
<td>۰</td>
<td>[-۲.۰۸۴,۲.۰۸۴]</td>
<td>۲</td>
<td>De Jong's F2</td>
<td>T2</td>
</tr>
<tr>
<td>۰</td>
<td>[-۸.۸]</td>
<td>۵</td>
<td>Ackey F1</td>
<td>T3</td>
</tr>
<tr>
<td>۰</td>
<td>[-۸.۳]</td>
<td>۱۰</td>
<td>Rastrigin F1</td>
<td>T4</td>
</tr>
<tr>
<td>۰</td>
<td>[-۵.۱۰]</td>
<td>۵</td>
<td>Rosenbrock</td>
<td>T5</td>
</tr>
</tbody>
</table>

جدول مسیرباین مورچه بر اساس مقادیر جدید فرومون و پارامتر مکانیافته به روزرسانی می‌شود.

در سورتی که حفرات محکه به هر پارامتر پیدا می‌کنند، از یک مدل کنتشار می‌شود تا به حفرات باز می‌ایستد به اصل می‌پردازد. در انتهای کار به روزرسانی برخی و با تأخیر فرومون انیور سیم، به این طریق که مقادیر فرومون مربوط به جهت را به پارامتر مکانیافته (ηi) مربوط به آن مکانیافته است. به مقادیر بیشتری افزایش می‌دهیم.

اگر از حافظه استفاده شده باشد، مقادیر فرومون جهت استفاده شده در حافظه به اساس معادله (۹) به روزرسانی می‌شود. سپس پارامتر اکشن مربوط به آن جهت (در سورتی که مقادیر چرخه کاهش پیدا کرد به جهت معین بهتر شده باشند) به اساس معادله (۱۰) به روزرسانی می‌شود.

\[ \tau_{new} = (1-\phi) \tau_{old} + \phi \tau_0 \]

\[ i \in \text{direction vector indices} \]

\[ \eta_i = |\text{f}_{current} - \text{f}_{previous}| \]

\[ i \in \text{direction vector indices} \]

اگر از حافظه استفاده نشده باشد، گام در سورتی که مقادیر پارامتر اکشن فعلی از همه پارامترهای اکتشافی استفاده شده در ساختن مسیرباین یک گروگان باشد (بینه جهت بهتری در رابطه ارزیابی داشته باشیم)، و همین طور مقادیر بحثی که بهترین نتایج داشته باشد (در بالمینه می‌زند). ان گام یک دقیقه به دست می‌آید. در این طور طول برداری بردار کرویانه محاسبه شده، نرمال می‌شود (تقسیم بر طول بردار می‌شود). سپس این بردار نرمال شده در حافظه به جای بدلخیانی نصیر ان که می‌گردد. همین طور مقادیر فرومون و پارامتر مکانیافته مربوط به برداری است که در حافظه ذخیره شده است. به روزرسانی می‌شود. در انتهای این مرحله مقداری

3۹

استقلال، سال ۱۲۵۱، شماره ۰، شماره ۰۱

۱۳۸۵
جدول 2- مقادیر پارامترها برای الگوریتم‌های مختلف

<table>
<thead>
<tr>
<th>GA</th>
<th>ES (μ + λ)</th>
<th>برنامه نویسی تکاملی</th>
<th>GCACO</th>
<th>الگوریتم</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>تعداد اجرا</td>
</tr>
<tr>
<td>300</td>
<td>150</td>
<td>150</td>
<td></td>
<td>تعداد نسل‌ها با اجتماع</td>
</tr>
<tr>
<td>400</td>
<td>T1-T3: μ = 0, λ = 100</td>
<td>T1-T3: μ = 50, λ = 100</td>
<td>T5: 60</td>
<td>اندازه جمعیت با تعداد</td>
</tr>
<tr>
<td></td>
<td>T4, T5: μ = 200, λ = 300</td>
<td>T4, T5: μ = 200, λ = 100</td>
<td></td>
<td>مورچه‌ها با (μ, λ)</td>
</tr>
<tr>
<td>0.01</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>نرخ ترکیب</td>
</tr>
<tr>
<td>0.2</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>نرخ جهش</td>
</tr>
</tbody>
</table>

جدول 3- مقادیر بهینه متغیرها و مقدار بهینه توایع T1, T2, T3 به ترتیب داده شده است

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>

جدول 4- مقادیر بهینه متغیرها و مقدار بهینه برای تایع T4

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>

بهرین جواب

جدول 5- مقادیر تابع

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>

جدول 6- مقادیر بهینه متغیرها و مقدار بهینه برای تایع T5

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>

جدول 7- مقادیر بهینه متغیرها و مقدار بهینه برای تایع T6

<table>
<thead>
<tr>
<th>T1</th>
<th>T2</th>
<th>T3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
</tr>
</tbody>
</table>


جهان مه‌کشین نتایج آزمایش‌ها برای توایع Rosenbrock، تعداد تکرار آن قابل تعیین شده می‌باشد. در اینجا به ذکر است که مقادیر بهینه سراسری همه توایع عناصر می‌باشد. مقادیر پارامترهای مشابه الگوریتم‌های تکاملی و الگوریتم پیشنهادی در جدول 2 داده شده است. نتایج آزمایش‌ها برای Rosenbrock و در جدول 3 داده شده است.

40

استقلال، سال 1385، شماره 1، شهریور 1202

Downloaded from iamt.utm.ac.ir at 16369 IRDT on Sunday September 20th 2020
شکل ۳- همگرایی الگوریتم GCACO در ۲۵ اجرا با تعداد ۵۰ اجتماع در هر اجرا، برای توابع T1, T2, T3, T4

جدول ۵- مقادیر بهینه متغیرها و مقادیر بهینه برای تابع Rosenbrock

<table>
<thead>
<tr>
<th>متغیر</th>
<th>مقادیر تابع</th>
<th>GCACO</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(۱)</td>
<td>۱۰۰۴</td>
<td>۱۰۰۴</td>
</tr>
<tr>
<td>W(۲)</td>
<td>۱۰۰۴</td>
<td>۱۰۰۴</td>
</tr>
<tr>
<td>W(۳)</td>
<td>۱۰۰۴</td>
<td>۱۰۰۴</td>
</tr>
<tr>
<td>W(۴)</td>
<td>۱۰۰۴</td>
<td>۱۰۰۴</td>
</tr>
</tbody>
</table>

استقلال، سال ۱۳۸۵، شماره ۱، شهریور ۱۳۸۵
4- نتیجه گیری

یک روش جدیدی با نام اختصار می‌باشدGCACO برای بهینه‌سازی

توابع در فضای یوپونه بر اساس الگوریتم بهینه‌سازی اجتماعی مورچگان ارائه شده است. اکرچه الگوریتم اجتماعی مورچگان برای بهینه‌سازی در فضای گسترشده و بهینه‌سازی ترکیبی نسبتاً پیشرفتی کاربردهای فراوانی پیدا کرده است، اما نوعی اصلاح این مقاله تعیین و توسعه الگوریتم به فضای یوپونه و به خصوص بهینه‌سازی نسبت به الگوریتم‌های قبلی بیشتر شبیه به مورچگان‌ها واقعی است. نتایج نشان می‌دهد که الگوریتمGCACO از نظر محاسباتی بسیار کاربردی از الگوریتم تکاملی است. اکرچه GCACO می‌تواند دیگر الگوریتم تکاملی اختراب به نظیر الگوریتمهای دارد اما تاکید به دست آمده برای مسائل محک‌محج نشان‌دهنده بی‌پایانی یک مسیر مطابق با ریسیدن به بهینه سراسری در مقایسه با سایر روشهای اینیهمگراپی مطلوب به علم

پارامترها سریعتر همگرا می‌شود.

در آزمون بعدی نش نظر علامت بردار درازه‌گردانی به جای خود آن بررسی شده است. به این طریق که به جای بردار نرمال شده گردانی بردار که شمار علامت آن است در حافظه ذخیره و به کار گیری شود

مقایسه شکل‌های (۵) و (۶) نشان می‌دهد که درنظرگرفتن علامت بردار گردانی به جای خود بردار گردانی می‌تواند در بهترین جواب را به وجود بیشتر

<table>
<thead>
<tr>
<th>الگوریتم</th>
<th>مقدار تابع</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZAD6</td>
<td>6/521</td>
</tr>
<tr>
<td>برنامه نویسی تکاملی</td>
<td>8/371</td>
</tr>
<tr>
<td>استراتژی تکاملی</td>
<td>0/139</td>
</tr>
<tr>
<td>GCACO</td>
<td>8/047</td>
</tr>
</tbody>
</table>

در GA و DE به بعد در دیل آزمون است، می‌توان نتیجه گرفت که در مسائل با ابعاد بالا از نظر یافتن جواب بهینه سراسری در این مورد خوب عمل می‌کند. در حالی که سایر الگوریتم‌های تکاملی و همین طور الگوریتم GCACO برای توانایی این مورد در مقاله با تعداد نسبتاً و جمعیت سایر الگوریتم‌ها کارکردها در مسائل مختلف بهینه‌سازی GCACO به گفته پیش‌بینی می‌توان استفاده کرد. همان طوری که از نتایج می‌توان استکبار کرد، به‌طور مثال:

| پرتره ندارد [۷۱] |}

در ادامه مطالب اینجا نشان می‌دهد، به عنوان اولیه T4 مورد بررسی قرار گرفت. اگرGCACO با مقدار پارامتر q = 1 اجرا شود، فقط حرکت تصادافی ملاحظه می‌شود و نش نظراتی می‌باشدGCACO تا جمعی (حاصل و فرآیند) حذف می‌شود. بنابراین با حذف نقش فرآیند و حاصله، الگوریتم بهینه‌سازی مانند یک الگوریتم جستجوی جهت نوریدی خاص عمل می‌کند. شکل (۵). همگراپی GCACO را با نظر گرفتن حافظه و فرآیند و GCACO به‌طور مثالی در نظر گرفتن این این دانسته می‌دهند. همان طوری که در شکل‌ها مشاهده می‌شود، نش نظراتی می‌باشدGCACO در اجتماعات ایالت بیشتر آشکار می‌شود (الگوریتم GCACO با وجود این ایجادل، سال ۱۲۵۷، شماره ۱، شریروز ۱۳۸۵

42
شکل 5- همگراىى گورینم GCACO در 25 اجرای با تعداد 50 اجتماع در هر اجرا (با وجود پارامترهای ارتباط با محیط و بدون آنها) 

شکل 6- همگراىى گورینم GCACO در 25 اجرای با تعداد 50 اجتماع در هر اجرا (با وجود پارامترهای ارتباط با محیط و بدون آنها). در این شکلها نحوه همگراىى گورینم با درنظر گرفتن علائم بردار گرافیان به جای بردار گرافیان تنشأ داده شده است.


