الگوریتم بهینه‌سازی اجتماع مورچگان بر مبنای گرادیان برای فضاهای پیوسته

مهدی افتخاری*، بیژن داعی** و سراج الدین کاتبی***
بخش مهندسی و علوم کامپیوتر، دانشکدهٔ مهندسی دانشگاه شیراز

(دریافت مقاله: ۸۳/۹/۱- دریافت نسخه نهایی: ۸٤/۱۲/۲۸)

چکیده – در این تحقیق یک نسخه جدید از الگوریتم بهینه‌سازی اجتماع مورچه‌ها[1] که توانایی جستجو در فضای پیوسته[2] را دارد، ارائه می‌شود. ساختار و مفاهیم اصلی الگوریتم اولیه بهینه‌سازی اجتماع مورچه حفظ شده وتعمیم و توسعه آن به فضای پیوسته انجام و پیاده‌سازی شده است. خاصیت ارتباط غیرمستقیم از طریق محیط (استیگمرجی[3]) با تعدادی بردار گرادیان نرمال شده شبیه‌سازی شد. برای اینکه همه مورچه‌ها بتوانند محیط را حس کنند، این بردارها توسط یک حافظه مشترک نگهداری می‌شوند. الگوریتم بهینه‌سازی پیشنهادی، بر روی توابع خاصی که به عنوان محک[4] در مسائل بهینه‌سازی فضای پیوسته به کار می‌روند، امتحان شده است. نتایج به دست آمده از این الگوریتم با نتایج الگوریتمهای تکاملی مانند الگوریتم ژنتیکی[5]، استراتژی تکاملی[6] و برنامه‌نویسی تکاملی[7] مقایسه شده و از لحاظ دقت وحجم محاسبات مورد نیاز نتایج حاصل از الگوریتم پیشنهادی به خوبی با الگوریتمهای دیگر قابل رقابت و در بعضی موارد بهتر است.

واژگان کلیدی: اجتماع مورچه، تکاملی، الگوریتمها، فوق اکتشافی، اتفاقی، محدودیت

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

---

* – دانشجوی دکترا ** – دانشجوی کارشناسی ارشد *** – استاد

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

and the results are compared and contrasted with other population-based algorithms such as Evolutionary Strategies (ES), Evolutionary Programming (EP), and Genetic Algorithms (GA). The results obtained from GCACO compare satisfactorily with those of other algorithms and in some cases are superior in terms of accuracy and computational demand.

Keywords: ***Ant Colony, Evolutionary, Algorithms, Meta-heuristic, Stochastic, Constraint.***

۱- مقدمه

الگوریتمهای بهینه‌سازی اجتماع مورچگان جزو دسته الگوریتمهای جستجوی فوق اکتشافی[۸] اند که از رفتارهای اجتماعی مورچه‌ها و موریانه‌های واقعی الهام گرفته شده‌اند. کاربرد متداول آنها در مسائل بهینه‌سازی با فضای گسسته (مانند مسئله معروف فروشنده دوره گرد[۹]) است. اجتماع واقعی مورچه‌ها را می‌توان به صورت یک سیستم توزیع شده بزرگ دید که می‌تواند مسائل را (از قبیل پیدا کردن منبع غذا) با استفاده از خاصیتی به نام استیگمرجی حل کند. تحریک کارگرها به کار بیشتر به علت کارایی که به دست می‌آورند، اولین بار توسط گراسه[۱۰] به عنوان استیگمرجی تعریف شد [۱]. این تعریف نوعی ارتباط غیرمستقیم بواسطه تغییرات محیط را بیان می‌کند [۲]. گراسه این ارتباط را در دو نوع خاص از موریانه‌ها به نامهای بلیکزیترمزناتالنسیس[۱۱] وکوبیترمز[۱۲] مشاهده کرد. خاصیت نشست فرومون[۱۳] قسمتی از فرایند سربازگیری[۱۴] است. فرایند سربازگیری توسط زیست شناسان نوعی سازماندهی خاص تعریف می‌شود، که در آن اعضای یک جامعه به سمت یک نقطه از فضای کاری جهت‌گیری می‌کنند [۳].

سیستم مورچه[۱۵] اولین الگوریتم بهینه‌سازی اجتماع مورچه بود که در سال ۱۹۹۱ [٤و٥] توسط دوریگو[۱۶] پیشنهاد شد. اولین بار سیستم مورچه توسط مسئله معروف فروشنده دوره گرد که مسئله بهینه‌سازی ترکیبی است، مورد آزمایش قرار گرفت. این الگوریتم حداکثر به تعداد T تکرار اجرا می‌شود. در هر تکرار m عدد مورچه یک تور[۱۷] که شامل n مرحله است را با استفاده از یک قانون تصمیم‌گیری مبتنی بر احتمال به دست می‌آورند.

در الگوریتم سیستم مورچه بعد از اینکه مورچه‌ها تور خود را کامل کردند (هر کدام جواب خود را به دست آوردند)، هر مورچه مقداری فرومون توسط متغیرهای مربوط بر روی یالهای ملاقات شده قرار می‌دهد تا این یالها توسط مورچه‌های بعدی بیشتر مورد توجه قرار گیرند، سپس مورچه‌ها می‌میرند. در کنار اضافه شدن فرومونها، مقداری از فرومونها تبخیر می‌شود که نقش اساسی در جلوگیری از رکود[۱۸] دارد. (رکود یا درجا زدن زمانی اتفاق می‌افتد که همه مورچه‌ها یک جواب به دست بیاورند).

اما حافظه یا حالت داخلی[۱۹] مورچه شامل نقاط ملاقات شده تا کنون است و لیست ممنوع[۲۰] نامیده می‌شود. این حافظه برای این به کار می‌رود که مورچه k ام با استفاده از آن بتواند بفهمد که از نقطه فعلی که در آن قرار دارد باید به کجا حرکت کند و چه نقطه‌هایی را مجبور است برای تکمیل تور خود طی کند.

در این الگوریتم یک جدول تصمیم[۲۱] یا جدول مسیریابی مورچه[۲۲] وجود دارد که مقادیر موجود در آن مبنای محاسبه احتمالات برای انتخاب جهتهای گوناگون است. این جدول با نماد A نمایش داده می‌شود.

$$A = [a_{ij}(t)] \qquad a_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in N_i}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} \quad \forall j \in N_i \qquad (۱)$$

در معادله (۱) η پارامتر اکتشاف[۲۳] است (در مورد مسئله فروشنده دوره گرد، $\eta_{ij} = 1/d_{ij}$، به معنی این است که مسافرت از نقطه i به j چقدر خوب است، جایی که d_{ij} فاصله بین نقاط i و j بوده و N_i مجموعه همسایه‌های گرهٔ i ام است.) و τ_{ij} مقدار فرومون بین دو گره i و j را نشان می‌دهد. پارامترهای $0 \le \alpha \le 1, 0 \le \beta \le 1$ توانهایی هستند که توسط آنها می‌توان میزان تأکید بر فرومون و پارامتر اکتشاف را تنظیم

کرد. حال در تکرار t احتمال این کـه مورچـه k ام نقطـه j ام را برای رفتن انتخاب کند، در حـالی کـه در نقطـه i ام اسـت، بـه صورت زیر است:

$$P_{ij}^{k}(t)=\frac{a_{ij}(t)}{\sum_{l\in N_i^k} a_{il}(t)} \qquad (۲)$$

N_i^k در معادله (۲) زیر مجموعه ای از N_i اسـت و شـامل نقاطی است که توسط مورچه k ام هنوز ملاقات نشده‌اند. (ایـن نقاط با استفاده از فهرست ممنوع قابل تشخیص‌اند).

اضافه شدن مقدار فرومـون و همچنـین تبخیـر فرومـون بـا معادله زیر در هرتکرار صورت می‌پذیرد.

$$\begin{cases} \Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \\ \tau_{ij}(t)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}(t) \end{cases} \qquad (۳)$$

در معادلـه بـالا ρ پارامترکـاهش فرومـون (تبخیـر) اسـت و $\Delta\tau_{ij}^{k}(t)$ مقدار فرومونی است که توسـط مورچـه k ام روی یال (i,j) نشست کرده اسـت و $\Delta\tau_{ij}(t)$ مقـدار کـل فرومـون نشستی روی یال (i,j) ام است.

الگوریتمهای سیستم مورچه با مـسایل فروشـنده دوره گـرد با اندازه کوچک (در حدود ۳۰ تا ۷۵ شهر)، بـا سـایر روشـهای اکتشافی مقایسه شده اسـت[٦]. طبـق گـزارش دوریگـو [٦-٨] نتایج به دست آمده از طرفـی خیلـی جالـب و از طـرف دیگـر خیلی ناامید کننده است. الگوریتم سیستم مورچه قـادر بـود کـه بهترین نتایج به دست آمده توسـط الگوریتمهـای ژنتیـک بـرای مسئله اولیور[۲٤]-۳۰ را بهبود بخشد و کـارایی شـبیه یـا بهتـر از الگوریتمهای اکتشافی داشته باشد. از طرفی برای مسایل با ابعاد بالا سیستم مورچه هرگز به بهترین جوابها نرسید [۸].

الگـوریتم سیـستم اجتمـاع مورچـه[۲۵] توسـط دوریگـو وگامباردلا[۲٦] [۹-۱۲] برای بهبود کارایی سیستم مورچه پیـشنهاد شد. الگوریتم سیستم اجتمـاع مورچـه بـا سیـستم مورچـه سـه تفاوت اساسی دارند که در [٦] به تفصیل توضیح داده می‌شـود. این الگوریتم، بهتـرین الگـوریتم بهینـه‌سـازی اجتمـاع مورچـه تاکنون است، که توانسته است نقایص مربوط بـه الگوریتمهـای سیستم مورچه را برطرف کرده و کارایی مطلوب را داشته باشد. از این مرحله به بعد هرجا عبارت الگوریتم بهینه‌سازی اجتمـاع مورچه استفاده شده است، منظور ساختار کلی الگوریتم سیـستم اجتماع مورچه است.

در زمینه تعمیم الگوریتمهای بهینه‌سازی اجتماع مورچه‌ها به فضای پیوسته تحقیقات اندکی صورت گرفته اسـت کـه اولـین آنهـا توسـط بیلچـف[۳۰] [۱۳] در سـال ۱۹۹۵ انجـام شـد. بقیـه روشهایی که پیشنهاد شده است، بـه ترتیـب در [١٤-١٦] آورده شده است. در اکثر این روشها جستجو در دو مرحلـه سراسـری و محلی انجـام مـی‌شـود، کـه مرحلـه سراسـری آن بیـشتر بـه الگوریتمهـای ژنتیـک شـبیه اسـت. در ایـن مقالـه، الگـوریتم پیشنهادی بر مبنای چهارچوب کلـی الگـوریتم سیـستم اجتمـاع مورچه است که به فضای پیوسته تعمیم داده شده است.

۲- روش پیشنهادی، تعمیم الگـوریتم بهینـه‌سـازی اجتماع مورچه به فضای پیوسته

در این بخش ما به توضیح الگوریتم پیشنهادی مـی‌پـردازیم. در شکل (۱) جریانی الگوریتم و شبه کـد بهینـه‌سـازی اجتمـاع مورچه پیشنهادی که به اختصار GCACO نامیده می‌شود، آورده شده است. سپس موارد تعمیم و اصلاح الگوریتم بـرای فـضای پیوسته شرح داده می‌شود.

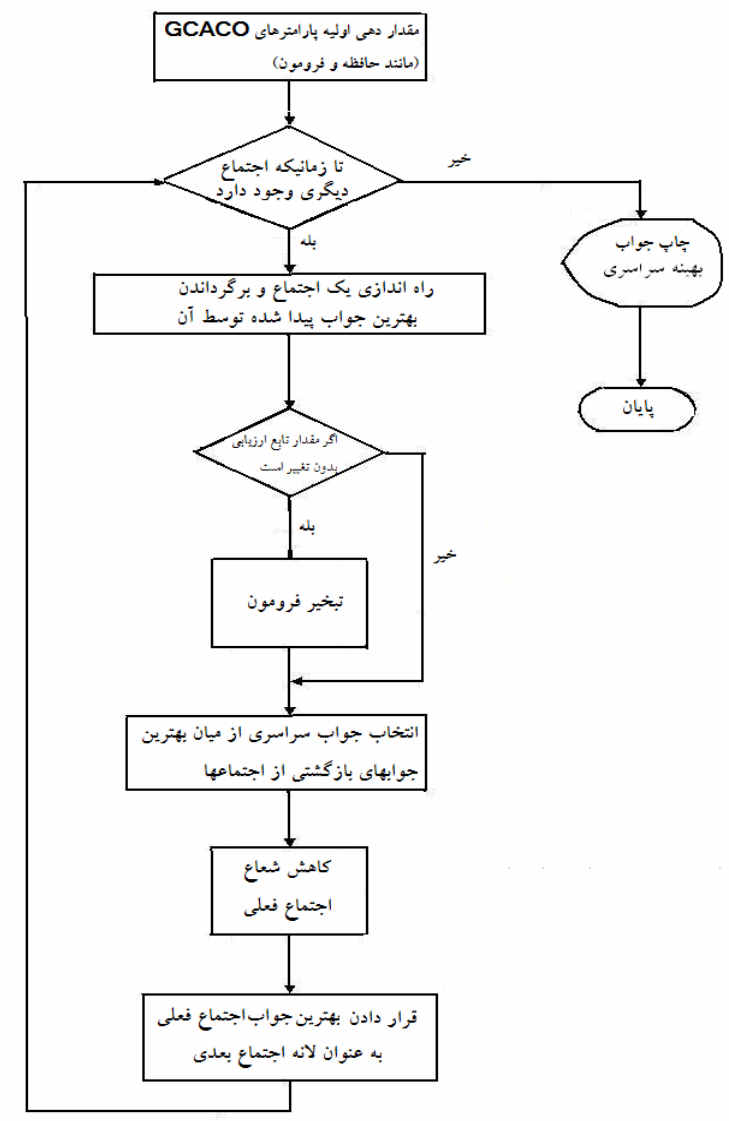
اولین تفاوت این است که در الگوریتم پیشنهادی بـرای هـر اجتماع یک شعاع حرکت $0 \le R_c \le 1$ در نظر گرفته شده است (برای اولین اجتماع مقدار شـعاع ۱ در نظـر گرفتـه مـی‌شـود). سپس مقدار شعاع توسط یک تابع نمایی به شـکل زیـر کـاهش پیدا می‌کند.

$$R_c = R_c.e^{\left(-t/T_{max}\right)} \qquad (٤)$$

T_{max} تعداد تکرارها، t تکرار فعلی است.

تفـاوت دیگـرGCACO بـا الگـوریتم بهینـه‌سـازی اجتمـاع مورچه معمولی این است که بهترین نقطه به دست آمده توسـط اجتماع فعلی بـه عنـوان لانـه[۳۱] بعدی درنظر گرفتـه مـی‌شـود.

[DOR: 20.1001.1.2251600.1385.25.1.3.3]



شکل ۱- نمودار جریانی مربوط به الگوریتم پیشنهادی

این خاصیت الگوریتم آن را به روشهای حریصانه[۳۲] جستجو مانند تپه نوردی[۳۳] نزدیک ساخته است.

تبخیر فرومون که درخط ٦ از شکل (۲) آمده است، توسط معادله زیر انجام می‌شود:

$$\tau^{new} = evap\_factor \times \tau^{old} \qquad (۵)$$

عمل تبخیر فرومون زمانی اتفاق می‌افتد که تابع ارزیابی[۳۴] تغییری پیدا نمی‌کند. (این به معنی آن است که الگوریتم احتمالاً دچار رکود شده است و فرومونهای قبلی باید به فراموشی سپرده شوند.)

با توجه به شکل(۲)، هر مورچه توسط تابع new_active_ant() به راه می‌افتد و یک جواب برمی‌گرداند که بهترین جوابی است که پیدا کرده است. سپس از بین این جوابها، بهترین جواب سراسری انتخاب می‌شود و به عنوان نتیجه تلاش اجتماع مورچگان تلقی می‌شود. هر مورچه توسط تابع new_active_ant() مقادیرمتغیرهای فرومون، حافظه و جدول مسیریابی را تغییر می‌دهد. این مقادیر برای حرکت مورچه بعدی استفاده می‌شود. فرق حافظه این الگوریتم با الگوریتم بهینه‌سازی اجتماع مورچه معمولی این است که در اینجا حافظه بین همه مورچه‌ها مانند فرومون و جدول مسیریابی مشترک است.

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

1 Procedure GCACO_Meta_heuristic( )
2 Initialize_the_parameters_of_GCACO ();
3 While (t < T_{max})
4 [Best_Eval_Function, Best_solution] = ants_generation_and_activity ();
5 if No_change_in_ Eval_Function
6 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 [best_fitness, best_location, routing_table, memory, τ]=new_active_ant();
18 select_global_location_among_the_best_ones;
19 set_returned_τ_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 q = Uniform_rand(0,1);
28 if $q \le q_0$
29 $x_{new} = x_{old} + R_a \times U(-1,1)$
30 else
31 best_index = compute_best_direction();
32 $x_{new} = x_{old} + R_a \times memory(best\_index)$
33 endif-else
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 ($\eta_{best\_index} < \eta_{current}$ and $f_{current} < f_{previous}$ **)**
38 $\eta_{best\_index} = \left| f_{current} - f_{previous} \right|$
39 endif
40 else
41 $\eta_{current} = \left| f_{current} - f_{previous} \right|$
42 if ($\forall i \quad \eta_i < \eta_{current}$ and $f_{current} < f_{previous}$ **)**
43 NGV = calculate_the_gradient_and_normalize_it;
44 Bad_index = 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 $\eta_{Bad\_index} = \left| f_{current} - f_{previous} \right|;$

48 endif
49 endelseif
50 Update_ant_routing_Table();
51 $f_{previous} = f_{current}$
52 $location_{previous} = location_{current}$
53 if (Ra < 0)
54 Active_ant = false;
55 endif
56 Ra = Ra * DF;
57 endwhile
58 if (online_delayed_pheromone_update)
59 Bh = Find_direction_with_best_heuristic_value;
60 $\tau_{Bh}^{new} = (1-\varphi) \times \tau_{Bh}^{old} + \varphi \times 4;$
61 Update_ant-routing_Table();
62 end if
63 endProcedure

GCACO شکل ۲- شبه کد الگوریتم بهینه‌سازی اجتماع مورچگان برای فضای پیوسته

در الگوریتم new_active_ant() تا زمانی که یک مورچه فعال است، حرکت را بر اساس یک قانون تصمیم‌گیری تصادفی که با الگوریتم بهینه‌سازی اجتماع مورچه معمولی فرق می‌کند، ادامه می‌دهد. حرکت مورچه‌ها در فضای جستجو کاملاً تصادفی یا بر اساس بردارهای گرادیان ذخیره شده در حافظه است. ابتدا یک عدد تصادفی بین [۱، ۰] تولید می‌شود. اگر این عدد از q. کوچکتر بود، مقدار مکان جدید به صورت $\overrightarrow{x_{new}} = \overrightarrow{x_{old}} + Ra.\overrightarrow{U(-1,1)}$ به دست می‌آید. این رابطه حرکت کاملاً تصادفی مورچه را شبیه‌سازی می‌کند. به این ترتیب که مورچه حرکت خود را با شعاع Ra و در امتداد یک بردار تصادفی که عناصر آن با استفاده از یک توزیع تصادفی یکنواخت بین [۱، ۱-] به دست می‌آیند، تغییر می‌دهد. در غیر این صورت حرکت مورچه بر اساس حافظه است. حافظه از تعداد محدودی جهت تشکیل شده است و در صورت لزوم، بهترین جهت بر اساس احتمال منسوب شده به هر جهت انتخاب می‌شود و حرکت بعدی مورچه بر اساس این جهت است. احتمال جهت i ام از معادله (۷) و بر اساس مقادیرموجود در جدول مسیریابی، معادله (٦)، به دست می‌آید. یک مکانیزم حریصانه یا انتخاب چرخشی مانند وسیله قمارچرخان[۳۵] برای انتخاب بهترین جهت می‌تواند استفاده شود.

$$a_i = \frac{\tau_i \times \eta_i^{\beta}}{\sum_{j=1}^{\text{Directions}} \tau_j \times \eta_j^{\beta}} \quad \forall i \in \{\text{Directions}\} \text{ where } \beta=0.4 \qquad (٦)$$

$$P_i = \frac{a_i}{\sum_{j=1}^{\text{Directions}} a_j} \qquad (٧)$$

حرکت بر مبنای حافظه توسط معادله (۸) مدل می‌شود. جهتهای موجود در حافظه بردارهای نرمال شده گرادیان‌اند که در مراحل قبلی الگوریتم باعث بهبود در تابع ارزیابی شده‌اند. Best_index اندیس بهترین جهت از جهتهای حافظه است که با توجه به احتمال جهتها و یک روش حریصانه یا چرخشی انتخاب می‌شود.

$$\overrightarrow{x_{new}} = \overrightarrow{x_{old}} + Ra.\overrightarrow{memory(best\_index)} \qquad (٨)$$

بعد از حرکت هر مورچه به مکان جدید، ممکن بودن[۳٦] آن مکان برای اینکه از قیود تعریف شده تجاوز نکند بررسی می‌شود.

جدول ۱ – جزییات توابع محک

شناسه	نام تابع	فرمول ریاضی	بعد	محدوده جستجو	بهینه سراسری
T1	Schaffer F6	$\min f(x,y)=0.5+\frac{(\sin^2(\sqrt{(x^2+y^2)})-0.5)}{((1+0.001\times(x^2+y^2))^2)}$	2**	[-100,100]	0
T2	De Jong's F2	$\min f(x,y)=100(x^2-y)^2+(1-x)^2$	2	[-2.084,2.084]	0
T3	Ackey F1	$\min f(x)=20+e-20\exp(-0.2\sqrt{0.2\sum_{i=1}^{5}x_i^2})-\exp(0.2\sum_{i=1}^{5}\cos(2\pi x_i))$	5	[-8,8]	0
T4	Rastrigin F1	$\min f(x)=100+\sum_{i=1}^{10}(x_i^2-10\cos(2\pi x_i))$	10	[-8,3]	0
T5	Rosenbrock	$\min R_n(X)=\sum_{i=1}^{n-1}\left[100(x_i^2-x_{i+1})^2+(x_i-1)^2\right]$	5,10	[-5,10]	0

اگر از حافظه استفاده شده باشد، مقدار فرومون جهت استفاده شده در حافظه بر اساس معادله (۹) به روزرسانی می‌شود، سپس پارامتر اکتشاف مربوط به آن جهت (در صورتی که مقدار تابع ارزیابی کاهش پیدا کرده باشد یعنی بهتر شده باشد)، بر اساس معادله (۱۰) به روزرسانی می‌شود.

$$\tau_i^{new}=(1-\varphi).\tau_i^{old}+\varphi.\tau_0 \qquad (۹)$$

$i \in$ direction vector indices

$$\eta_i=\left|f_{current}-f_{previous}\right| \qquad (۱۰)$$

$i \in$ direction vector indices

اگر از حافظه استفاده نشده باشد، آن گاه در صورتی که مقدار پارامتر اکتشاف فعلی از همه پارامترهای اکتشاف استفاده شده در ساختن جدول مسیریابی بزرگتر باشد (یعنی جهش بهتری درتابع ارزیابی داشته باشیم)، و همین طور مقدارتابع ارزیابی کاهش پیدا کرده باشد (در مسائل کمینه‌سازی)، آن گاه بردار گرادیان محاسبه شده، نرمال می‌شود (تقسیم بر طول بردار می‌شود). سپس این بردار نرمال شده در حافظه به جای بدترین عنصر آن قرار می‌گیرد. همین طور مقدار فرومون و پارامتر مکاشفه‌ای مربوط به برداری است که در حافظه ذخیره شده است، به روزرسانی می‌شود. در انتهای این مرحله مقادیر جدول مسیریابی مورچه بر اساس مقادیر جدید فرومون و پارامتر مکاشفه‌ای به روزرسانی می‌شوند.

در صورتی که شعاع حرکت مورچه که هر بار کاهش پیدا می‌کند، از یک حد کمتر شد، مورچه از حرکت باز می‌ایستد یا به اصطلاح می‌میرد. در انتهای کار به روزرسانی برخط و با تاخیر فرومون [۳۷] انجام می‌گیرد، به این طریق که مقدار فرومون مربوط به جهتی را که مقدار پارامتر مکاشفه‌ای (η) مربوط به آن ماکزیمم است، به مقدار بیشتری افزایش می‌دهیم.

۳- نتایج شبیه سازی

نتایج حاصل از الگوریتم پیشنهادی با نتایج حاصل از برخی الگوریتمهای تکاملی [۱۷] که مبتنی‌بر جمعیت جواب‌اند، مانند الگوریتم استراتژی تکاملی و برنامه‌نویسی تکاملی مقایسه شده است. در این مقایسه از پنج تابع معروف محک استفاده شده و خصوصیات آنها در جدول (۱) آورده شده است. این توابع محک معمولاً برای آزمایش کارایی الگوریتمهای تکاملی استفاده می‌شوند [۱۷و۱۸]. همچنین معیار توقف الگوریتمهای تکاملی ذکر شده، کوچکتر شدن مقدار تابع از یک مقدار بسیار کوچک از قبل تعریف شده و در صورت ارضا نشدن این معیار، اتمام آن

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

جدول ۲- مقادیر پارامترها برای الگوریتمهای مختلف

الگوریتم	GCACO	برنامه نویسی تکاملی	ES $(\mu+\lambda)$	GA
تعداد اجرا	۲۵	۲۵	25	۲۵
تعداد نسل[۳۸]ها یا اجتماع	T1-T4: ۵۰ T5:۶۰	۱۵۰	150	۳۰۰
اندازهٔ جمعیت[۳۹] یا تعداد مورچه‌ها یا (μ,λ)	۵۰	T1-T3 : μ = ۵۰ T4, T5 : μ = ۲۰۰	T1-T3 : μ =۵۰, λ=۱۰۰ T4, T5 : μ =۲۰۰, λ=۳۰۰	۴۰۰
نرخ ترکیب[۴۰]	--	--	۰	۰/۷
نرخ جهش[۴۱]	--	--	۱	۰/۲

جدول ۳- مقادیر بهینه متغیرها و مقدار بهینه توابع T1,T2,T3 به ترتیب داده شده است

الگوریتم	(بهترین جواب ، [مقدار تاریخ])		
	T1	T2	T3
ژنتیک	([-۰/۶۶۹, -۳/۰۷], ۰/۰۱)	([۰/۹۲۹,۰/۸۶۲], 5E-۳)	([-۵/۰۹۹E-۵, -۱/۷۵۱E-۴, -۷/۸۳۲E-۵, - ۱/۵۰۵E-۴, -۳/۷۱۸E-۶] , ۴/۴۶E-۴)
برنامه نویسی تکاملی	([-۱/۴۶, –۲/۷۸],۹/۷۲E-۳)	([۱/۰۰۱, ۱/۰۰۲], ۱/۳E-۵)	([۰/۰۲۲, ۰/۰۰۸, ۰/۰۱۳, ۰/۰۱۴, -۰/۰۰۴],۰/۰۶۶۱)
استراتژی تکاملی	([-۳/۱۱۹, -۰/۳۴۲] ,۹/۷۲ E-۳)	([۱/۰۰, ۱/۰۰], ۰)	([۱/۲۵۶E-۶, ۴/۵۰۸E-۷, ۶/۰۴E-۹, -۲/۹۶۹E-۷, –۷/۸۸۷E-۷], ۱/۹۱E-۶)
GCACO	(۱/۰۱E-۳×[-۰/۲۲۴۳, -۰/۱۴۷۳], ۷/۲۱E-۸)	([۱/۰۰۱,۱/۰۰۲], ۹/۱۵۸E-۷)	([۶/۳۹E-۶, ۵/۸۸E-۶, ۴/۰۱E-۶, -۱/۸۸E-۶, ۱/۲۴۷E-۵], ۲/۸۳۲E-۵)

جدول ۴- مقادیر بهینه متغیرها و مقدار بهینه برای تابع T4

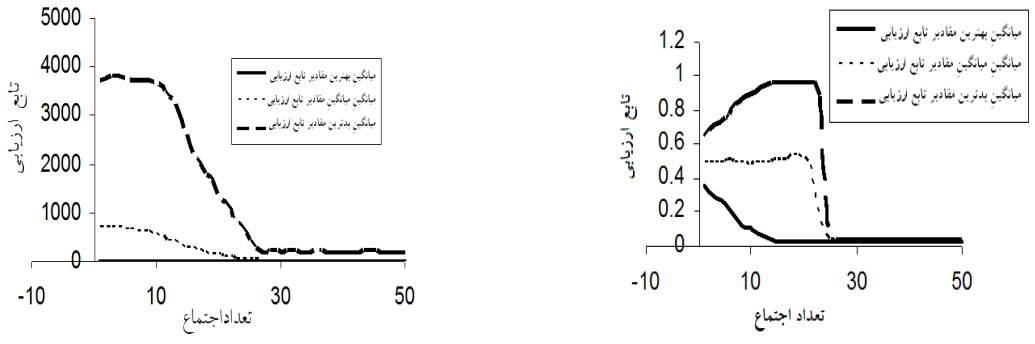
الگوریتم	مقدار تابع	بهترین جواب
ژنتیک	۶/۹۰۳E-۴	[۰/۰۰۱, ۸/۵۵E-۵, -۰/۰۰۱۳, ۱/۰۲۵E-۴, -۲/۴۸۹E-۴, -۲/۵۰۳E-۴, -۲/۹۰۷E-۴, -۱/۸۴۴E-۴, -۱/۹۸۳E-۴, ۲/۳۳۴E-۵]
برنامه نویسی تکاملی	۲۰/۸۷۷	[۰/۰۷۳۷, -۱/۰۵۴, ۰/۰۱۴۳, ۰/۹۵۱۸, -۰/۹۶۳۹, -۰/۹۶۷۴, -۱/۸۰۱۴, ۰/۱۲۱۸, ۰/۰۲۲۹,۰/۰۸۵۹]
استراتژی تکاملی	۱۱/۹۵۷	[۰/۹۹۵۶, ۰/۰۰۱۲, ۲/۵۶۱۲E-۴, -۰/۹۸۹۱, ۰/۹۹۵۷,۱/۹۹۱ , ۰/۰۰۲۶۴, ۰/۰۰۵۳۹, -۰/۹۹۱۵,۱/۹۹۱]
GCACO	۳/۵۸۲	[۰/۰۰۸, -۰/۰۰۱, -۰/۰۳۸, -۰/۹۸۳, ۰/۰۲۲, ۰/۹۹۸, ۰, ۱/۰۰۳, ۰/۰۱۱۳, -۰/۰۲۷]

تعداد تکرار از قبل تعیین شده می باشد. لازم به ذکر است که مقدار کمینه سراسری همه توابع محک صفر می‌باشد.

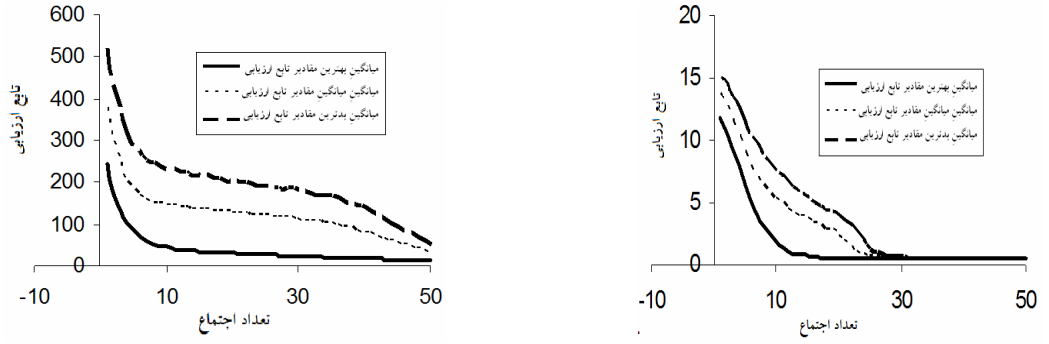
مقادیر پارامترهای مشابه الگوریتمهای تکاملی و الگوریتم پیشنهادی در جدول (۲) داده شده است. نتایج آزمایشها برای محک T1 تا T4 در جدولهای (۳ و ۴) آورده شده است. همچنین نتایج آزمایشها برای توابع Rosenbrock پنج بعدی و ده بعدی در جدولهای (۵ و ۶) آمده است.

همچنین نتایج آزمایشها برای توابع Rosenbrock پنج بعدی

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

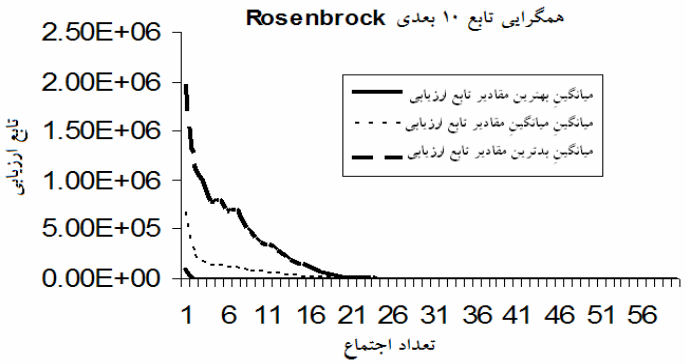


(الف) همگرایی T1 (ب) همگرایی T2



(ج) همگرایی T3 (د) همگرایی T4

شکل ۳- همگرایی الگوریتم GCACO در ۲۵ اجرا با تعداد ۵۰ اجتماع در هر اجرا، برای توابع T1,T2,T3,T4



شکل ۴- همگرایی الگوریتم GCACO در ۲۵ اجرا با تعداد ۶۰ اجتماع در هر اجرا

جدول ۵ – مقادیر بهینه متغیرها و مقدار بهینه برای تابع Rosenbrock پنج بعدی

الگوریتم	مقدار تابع	بهترین جواب (n=۵)
ژنتیک	۰/۰۱۸۶	[۱/۰۰۷, ۱/۰۰۲, ۱/۰۰۳, ۱/۰۰۶,۱/۰۰۵]
برنامه نویسی تکاملی	۲/۸۹۶E-٤	[۱/۰۰۱,۱/۰۰۱ , ۱/۰۰۰,۱/۰۰۱ ,۱/۰۰۰۲]
استراتژی تکاملی	۲/٤۱۵E-٦	[۱/۰۰۰۱, ۱/۰۰۰۱, ۱, ۱,۱/۰۰۰۱]
GCACO	۹/۷۹٤E-٤	[۰/۹۹۷۰, ۰/۹۹٤۰, ۰/۹۸۸۰, ۰/۹۷۸٦,۰/۹۵۵۱]

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

جدول ۶- مقادیر بهینه متغیرها و مقدار بهینه برای تابع Rosenbrock ده بعدی

الگوریتم	مقدار تابع	بهترین جواب (n=۱۰)
ژنتیک	٦/٥٤٨E-٦	[۰/۹۹۹۹,۰/۹۹۹۹ , ۰/۹۹۹۹, ۰/۹۹۹۹, ۰/۹۹۹۹,۰/۹۹۹۹ ,۹۹۹۹ , ۰/۹۹۹۹, ۰/۹۹۹۲, ۰/۹۹۹۹, ۰/۹۹۹۹]
برنامه نویسی تکاملی	۸/۳۷۱	[۰/٦٧٤٠,۰/٥٥٥٤ ,۰/٦٩٩٣, ۰/٧٩٩٢, ۰/٨٩٣٢, ۰/٩٤٤٣, ۰/٩٧١٨, ۰/٩٨٥٨, ۰/٩٩٢٩, ۰/٩٩٥٢]
استراتژی تکاملی	۰/۱۰۳۹	[۱/۰۰۹,۱/۰۰ , ۱/۰۰۹,۱/۰۰۷, ۱/۰۰۷, ۱/۰۰۲, ۱/۰۰۹, ۱/۰۰۲, ۱/۰۰۳, ۱/۰۰۹]
GCACO	۰/۰۹٤۷	[۰/۹۸۳۵,۰/۹۸۳۳, ۰/۹۸۸۸, ۰/۹۸۷۷, ۰/۹۹۷۷, ۰/۹۹۸۳, ۰/۹۹۹٥, ۰/۹۹۸۹, ۰/۹۹۲۹, ۰/۹۹٥۰]

و ده بعدی در ذیل آمده است. می‌توان نتیجه گرفت که GA 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 توابع با ابعاد بالا (در این مثال n=10) مشابه به یکدیگر عمل می‌کنند. اما به علت حجم محاسباتی کمتر، (تعداد کمتر اجتماعها و مورچه‌های این الگوریتم در مقایسه با تعداد نسلها و جمعیت سایر الگوریتمها) کاراتر بودن الگوریتم GCACO در مسائل مختلف بهینه‌سازی فضای پیوسته کاملاً مشهود است. همان طوری که از نتایج می‌توان استنباط کرد، GCACO از لحاظ کیفیت جواب در بیشتر موارد بهتر از ES و EP عمل کرده است. در هر حال بر اساس قضیه معروف و اثبات شده تحت عنوان no free lunch theorem هیچ کدام از روشها به تنهایی برای همه مسائل برتری ندارد [۱۷].

در ادامه مطالعات انجام شده، نقش خاصیت ارتباط مورچه‌ها توسط محیط در الگوریتم پیشنهادی با استفاده از تابع T4 مورد بررسی قرار گرفت. اگر GCACO با مقدار پارامتر q. =۱ اجرا شود، فقط حرکت تصادفی ملاحظه می‌شود و نقش پارامترهای تلاش دسته جمعی(حافظه و فرومون) حذف می‌شود. بنابراین با حذف نقش فرومون و حافظه، الگوریتم پیشنهادی مانند یک الگوریتم جستجوی تپه نوردی خاص عمل می‌کند. شکل(۵)، همگرایی GCACO را با در نظر گرفتن حافظه و فرومون و بدون در نظر گرفتن این دو نشان می‌دهند. همان طوری که در شکلها مشاهده می‌شود، نقش پارامترهای تلاش دسته جمعی در اجتماعات آخر بیشتر آشکار می‌شود (الگوریتم با وجود این پارامترها سریعتر همگرا می‌شود).

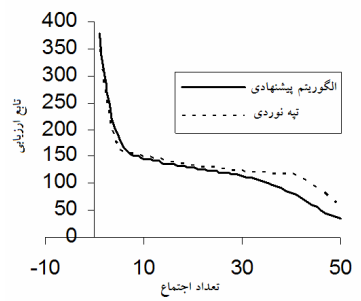
در آزمون بعدی نقش علامت بردار نرمال شده گرادیان به جای خود آن بررسی شده است. به این طریق که به جای بردار نرمال شده گرادیان برداری که شامل علامت آن است در حافظه ذخیره و به کارگیری شود.

مقایسه شکلهای (۵) و (٦) نشان می‌دهد که درنظرگرفتن علامت بردار گرادیان به جای خود بردار گرادیان می‌تواند در بهترین جواب را بهبود ببخشد.

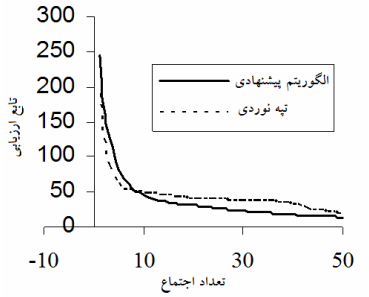
٤- نتیجه‌گیری

یک روش جدید با نام اختصار GCACO برای بهینه‌سازی توابع در فضای پیوسته بر اساس الگوریتم بهینه‌سازی اجتماع مورچگان ارائه شده است. اگر چه الگوریتم اجتماع مورچگان برای بهینه‌سازی در فضای گسسته و بهینه‌سازی ترکیبی نسبتاً پیشرفته کاربردهای فراوان پیدا کرده است، اما نوع‌آوری اصلی این مقاله تعمیم و توسعه این الگوریتم به فضای پیوسته و به خصوص بهینه‌سازی توابع است. رفتار الگوریتم پیشنهادی نسبت به الگوریتمهای قبلی بیشتر شبیه به مورچه‌های واقعی است. نتایج نشان می‌دهند که الگوریتم GCACO از نظر محاسباتی بسیار کاراتر از الگوریتمهای دیگر تکاملی مانند استراتژی تکاملی و برنامه‌نویسی تکاملی است. اگر چه GCACO همانند دیگر الگوریتمهای تکاملی احتیاج به تنظیم پارامترها دارد اما نتایج به دست آمده برای مسائل محک نشاندهندهٔ پیمایش یک مسیر مطلوب تا رسیدن به بهینه سراسری در مقایسه با سایر روشهاست. این همگرایی مطلوب به علت

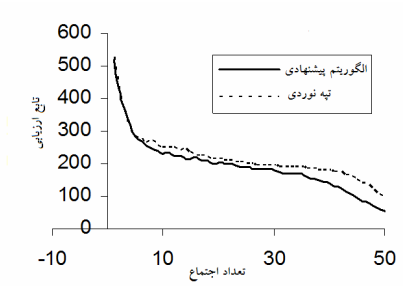
[DOR: 20.1001.1.2251600.1385.25.1.3.3]



الف) میانگین میانگین مقادیر تابع ارزیابی T4

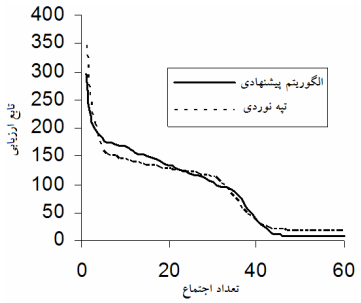


ب) میانگین بهترین مقادیر تابع ارزیابی T4

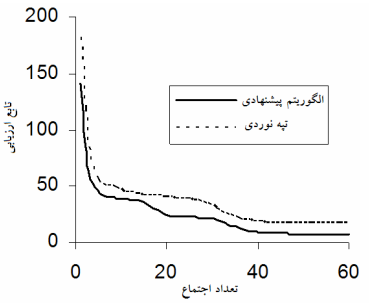


ج) میانگین بدترین مقادیر تابع ارزیابی T4

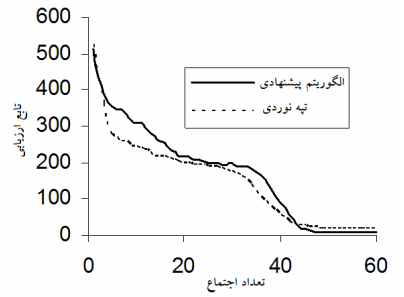
شکل ۵- همگرایی الگوریتم GCACO در ۲۵ اجرا با تعداد ۵۰ اجتماع در هر اجرا (با وجود پارامترهای ارتباط با محیط و بدون آنها)



الف) میانگین میانگین مقادیر تابع ارزیابی T4



ب) میانگین بهترین مقادیر تابع ارزیابی T4



ج) میانگین بدترین مقادیر تابع ارزیابی T4

شکل ۶- همگرایی الگوریتم GCACO در ۲۵ اجرا با تعداد ۵۰ اجتماع در هر اجرا (با وجود پارامترهای ارتباط با محیط و بدون آنها). در این شکلها نحوه همگرایی الگوریتم با درنظر گرفتن علامت بردار گرادیان به جای خود بردار گرادیان نشان داده شده است.

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

برقراری یک تعادل مناسب بین اکتشاف و بهره برداری در فضای جستجو هست که از طریق تنظیم مناسب پارامترها برای هر مسئله حاصل می‌شود. در این راستا شناخت مسئله و تجربه کاربر می‌تواند مفید واقع شود. آزمایشات انجام شده به خوبی نشان می‌دهند که اندازه جمعیت و تعداد نسلها در الگوریتم نسبت به سایر الگوریتمها کمتر است. این بدان معنی است که این الگوریتم از نظر محاسباتی کاراتر است. تسهیل در پیاده‌سازی، کارایی محاسباتی و نرخ همگرایی مطلوب از مزایای دیگر الگوریتم پیشنهادی است. اگر چه آثار و تبعات قضیه معروف no free lunch theorem (که بیانگر این است که هیچ کدام از روشهای حل یک مسئله برای تمامی مسائل مشابه کارا نیست) از آزمایشات انجام شده کاملاً مشهود است. اما کارایی محاسباتی الگوریتم پیشنهادی، حداقل در پنج مسئله بررسی شده به وضوح مشخص است.

واژه نامه

1. ant colony optimization
2. continuous spaces
3. stigmergic property
4. bench mark
5. genetic algorithm
6. evolutionary strategy
7. evolutionary programming
8. meta-heuristic
9. traveling sales man problem (TSP)
10. Grassé
11. Bellicositermes Natalensis
12. Cubitermes
13. pheromone deposition
14. recruitment
15. ant system
16. Dorigo
17. tour
18. stagnation
19. internal state
20. Tabu list
21. decision table
22. ant routing table
23. heuristic
24. Oliver-30
25. ant colony system (ACS)
26. Gambardella
27. pseudo random proportional rule
28. exploitation
29. bias exploration
30. Bilchev
31. nest
32. greedy
33. hill climbing
34. evaluation function
35. roulette wheel
36. feasibility
37. delayed online pheromone updating
38. generation
39. population size
40. combination (x-over)
41. mutation

مراجع

1. Grassé P. P., "La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. La théorie de la stigmergie: essai d'interprétation du comportement des termites constructeurs," *Insectes Sociaux,* Vol. 6, pp. 41–81,1995.
2. Theraulaz, G., and Bonabeau, E., "A Brief History of Stigmergy," *Artificial Life,* Vol. 5, pp. 97–116, 1999.
3. Pasteels, J. M., Deneubourg, J. L., and Goss, S., "Self-Organization Mechanisms in Ant Societies (i): Trail Recruitment to Newly Discovered Food Sources," *Experientia Supplementum*, Vol. 54, pp. 155-175, 1987.
4. Dorigo, M., "Optimization, Learning and Natural Algorithms (in Italian)," PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, 1992.
5. Dorigo, M., Maniezzo, V., and Colorni, A., "Positive Feedback as a Search Strategy," *Technical Report* 91-016, Dipartimento di Elettronica, Politecnico di Milano, IT, 1991.
6. Dorigo, M., Di Caro, G., and Gambardella, L. M., "Ant Algorithms for Discrete Optimization," *Artificial Life*, Vol. 5, No. 3, pp. 137-172, 1999.
7. Dorigo, M., Maniezzo, V., and Colorni, A., "The Ant System: Optimization by a Colony of Cooperating Agents,". *IEEE Transactions on Systems*, Man, and Cybernetics–Part B, Vol. 26, No. 1, pp. 29–41, 1996.
8. Whitley, D., Starkweather, T., and Fuquay, D., "Scheduling Problems and Travelling Salesman: The Genetic Edge Recombination Operator," *In Proceedings of the Third International Conference on Genetic Algorithms*, pp. 133–140. Palo Alto, CA: Morgan Kaufmann, 1989.
9. Dorigo M. and Gambardella, L. M., "Ant Colonies for the Traveling Salesman Problem," *BioSystems*, Vol. 43, pp. 73–81, 1997.
10. Dorigo M., and Gambardella, L. M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,". *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53–66, 1997.
11. Gambardella, L. M. and Dorigo, M., "Solving Symmetric and Asymmetric TSPs by Ant Colonies,"

[DOR: 20.1001.1.2251600.1385.25.1.3.3]

In Proceedings of the IEEE Conference on Evolutionary Computation, ICEC96, pp. 622–627, IEEE Press 1996.
12. Gambardella, L. M., and Dorigo, M, HAS-SOP: "A Hybrid Ant System for the Sequential Ordering Problem," *Technical Report* PP. 11-97, IDSIA, Lugano, CH, 1997.
13. BILCHEV, G., and PARMEE, I. C., "The Ant Colony Metaphor for Searching Continuous Design Spaces," *Proceedings of the AISB Workshop on Evolutionary Computation*, University of Sheffield, UK, April, pp. 3-4, 1995.
14. Mathur, M., Karale, S. B., Priye, S., Jyaraman, V. K. and Kullkarni, B. D., "Ant Colony Approach to Continuous Function Optimization," *Ind. Eng. Chem. Res.* Vol. 39, pp. 3814-3822, 2000.
15. Dréo, J., and Siarry, P., "A New Ant Colony Algorithm Using the Heterarchical Concept Aimed at Optimization of Multiminima Continuous Functions," *Proceedings of Third International Workshop, ANTS 2002*, Brussels, Belgium, 2002.
16. Li Yan-jun, Wu Tie-jun, "An Adaptive Ant Colony System Algorithm for Continuous-Space Optimization Problems," *Journal of Zhejiang University Science*, Vol. 4, No. 1, pp. 40-46, 2003.
17. Bach, T., Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. New York Oxford: Oxford University Press, 1996.
18. Fogel, D. B., Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. 2nd ed, New York: IEEE Press, 2000.

[DOR: 20.1001.1.2251600.1385.25.1.3.3]