مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir
دانشگاه کاشان
University of Kashan

حل مسئلهٔ در مدار قرار دادن نیروگاه‌ها با استفاده از الگوریتم رقابت استعماری اصلاح‌شده✧

سید عباس طاهر[1*]، استاد، مهدی حیدریان[2]، کارشناسی ارشد، احسان همنشین بهبهانی[3]، کارشناسی ارشد
[1و2و3] دانشکده مهندسی برق و کامپیوتر، دانشگاه کاشان، کاشان، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۰۵ اسفند ماه ۱۳۹۸
پذیرش ۱۲ خرداد ماه ۱۳۹۹

کلمات کلیدی:
در مدار قرار دادن نیروگاه‌ها
روش‌های تکاملی
الگوریتم رقابت استعماری اصلاح‌شده

چکیده

در مدار قرار دادن نیروگاه‌ها یکی از مهم‌ترین مسائل در بهره‌برداری از سیستم‌های قدرت است که در آن قیود مختلفی وجود دارند که باید رعایت شوند. این مسئله، غیرخطی و گسترده می‌باشد؛ به همین دلیل استفاده از الگوریتم‌های هوشمند برای حل آن بسیار مورد توجه قرار گرفته است. در این مقاله با بهره‌گیری از الگوریتم هوشمند رقابت استعماری اصلاح‌شده، که روش جدیدی است، حل مسئلهٔ در مدار قرار دادن ژنراتورها و تخصیص تولید واحدهای مختلف برای تأمین انرژی در طول برنامهٔ یک‌روزه در سیستم استاندارد ۱۰، ۶۰ و ۱۰۰ واحدی IEEE انجام گرفته است. نتایج به‌دست‌آمده نشان‌دهندهٔ صرفه‌جویی اقتصادی بیشتر نسبت به سایر الگوریتم‌های هوشمند مانند الگوریتم ژنتیک (GA)، الگوریتم بهبودیافتهٔ ژنتیک (ICGA)، الگوریتم اجتماع پرندگان (PSO) و بهبودیافته‌های آن و الگوریتم جست‌وجوی پرندهٔ فاخته (Cockoo Searching) است.

© ۱۳۹۹ ـ مجلهٔ محاسبات نرم، کلیه حقوق محفوظ است.

۱. مقدمه

یکی از مسائل بسیار مهم در کارکرد سیستم‌های قدرت، مسئلهٔ تخصیص واحدها (UC)[1] است. تخصیص واحدها یک مسئلهٔ غیرخطی، پیچیده و با ابعاد بزرگ برای تعیین برنامه‌ریزی روشن و خاموش شدن واحدهای تولیدی با کمترین هزینه است. در این مسئله تقاضا و رزرو آن باید تأمین و ثوابت واحدهای تولیدی نیز ارضا شوند [۱ـ۲]. این نکته نیز باید مدنظر قرار گیرد که حل مسئله می‌تواند در سیستم‌های قدرت بزرگ پرهزینه باشد، بنابراین راه‌حل‌های مختلفی برای حل این مسئله با کمترین هزینهٔ ممکن ارائه شده است که در یک دسته‌بندی کلی به دو بخش تقسیم می‌شوند [۳]. دستهٔ اول شامل تکنیک‌های بهینه‌سازی عددی مانند لیست حق تقدم [۴ـ۵]، برنامه‌ریزی پویا [۶ـ۷]، روش سینوس-کسینوس [۸] و روش لاگرانژ [۹ـ۱۱] بوده که روش‌هایی سریع و ساده هستند، اما بیشتر آن‌ها از محاسبات عددی گسترده و مشکلات محاسباتی رنج می‌برند [۱]. دستهٔ دوم شامل روش‌های مبتنی بر جست‌وجوی هوشمند مانند الگوریتم ژنتیک [۱۲ـ۱۴]، الگوریتم بهینه‌سازی اجتماع پرندگان (PSO) [۱۵] به‌همراه بهبودیافتهٔ آن [۱۶] و دولایهٔ آن [۱۷]، الگوریتم کلونی مورچه‌ها [۱۸] و الگوریتم جست‌وجوی پرندهٔ فاخته [۱۹]

✧ نوع مقاله: پژوهشی
* نویسنده مسئول
پست‌های الکترونیک: staher@kashanu.ac.ir (طاهر)
mheidarian@grad.kashanu.ac.ir (حیدریان)
eh.ha.be@gmail.com (همنشین بهبهانی)

1. Unit Commitment

نحوه ارجاع به مقاله: طاهر، سید عباس، حیدریان، مهدی، همنشین بهبهانی، احسان، «حل مسئلهٔ در مدار قرار دادن نیروگاه‌ها با استفاده از الگوریتم رقابت استعماری اصلاح‌شده»، مجله محاسبات نرم، جلد ۷، شماره ۲، ص۶۵ـ۷۳، پاییز و زمستان ۱۳۹۷.

هستند. این روش‌ها می‌توانند با موفقیت، شرایط غیرخطی و پیچیدهٔ حل مسئله را برآورده سازند و راه‌حل‌های نزدیک‌تر به نقطهٔ بهینه را فراهم سازند، اما همهٔ آن‌ها از ابعاد گسترده رنج می‌برند [۱]. در مرجع [۲۰] روش دیگری با عنوان روش مستقیم رو به جلو (SF[1]) ارائه شده است که مسئلهٔ UC را به سه زیر مسئلهٔ کوچک‌تر تقسیم می‌کند. این روش بر اساس خطی‌سازی تابع هزینهٔ واحدهای تولیدی بنا شده است.

آتش‌پز و لوکاس اولین معرفی‌کنندگان الگوریتم رقابت استعماری در سال ۲۰۰۷ بوده‌اند [۲۱] که بر مبنای رقابت‌های استعمارگرانه بنا شده است. در این روش تمام کشورها به دو دستهٔ استعمارگر و مستعمره تقسیم می‌شوند و مستعمره‌ها بین استعمارگران تقسیم می‌شوند و با هم دیگر امپراطوری‌هایی را تشکیل می‌دهند و سپس رقابت‌های استعمارگرانه‌ای بین امپراطوری‌ها شروع می‌شود. در پایان رقابت‌ها، تنها یک امپراطوری باقی می‌ماند که مستعمره‌های این امپراطوری همه در یک جایگاه هستند و ارزش یکسانی دارند [۲۲]. به‌تازگی این الگوریتم با موفقیت برای حل بعضی از مسائل بهینه‌سازی اعمال شده است [۲۲ـ۲۳].

در این مقاله بر اساس الگوریتم رقابت استعماری اصلاح‌شده بر مبنای لیست حق تقدم به حل مسئلهٔ در مدار قرار دادن نیروگاه‌ها پرداخته شده است. برای بررسی عملکرد، الگوریتم بر روی سیستم تست ۱۰، ۶۰ و ۱۰۰ واحدی استاندارد IEEE اعمال شده و شبیه‌سازی صورت گرفته است. نتایج نشان می‌دهند که تمامی قیود مطروح ارضا شده و جواب‌های حاصل با توجه به مقایسهٔ انجام‌گرفته با مراجع [۲، ۸، ۱۳، ۱۷، ۱۹، ۲۴ـ۲۶] به نقطهٔ بهینهٔ کلی نزدیک‌ترند.

۲. فرموله کردن مسئلهٔ UC

هدف در مسئلهٔ UC حداقل‌سازی هزینهٔ بهره‌برداری از واحدهای تولیدی در طول ساعات برنامه‌ریزی‌شده با توجه به رعایت قیود است. هزینهٔ بهره‌برداری شامل مجموع هزینه‌های راه‌اندازی، سوخت و هزینهٔ خاموش شدن واحدهای تولیدی است. توان خروجی واحدها در هر ساعت با پخش بار اقتصادی تعیین می‌شود. به‌طور کلی هزینهٔ سوخت با استفاده از تابع هزینهٔ سوخت واحدها به‌صورت زیر محاسبه می‌شود [۲۶]:

$$FC_i(P_i) = a_i + b_i P_i + c_i P_i^2 \quad i = 1,2,...,N \qquad (۱)$$

که a، b و cضرایب هزینهٔ سوخت واحدها، P_iتوان مورد تقاضا، i شماره واحد، $FC_i(P_i)$هزینهٔ واحد iام برای تولید توان مورد تقاضا هستند. بنابراین هزینهٔ نهایی سوخت برای تامین تقاضا در H ساعت به‌صورت زیر خواهد بود [۲۶]:

$$TFC = \sum_{t=1}^{H}\sum_{i=1}^{N} FC_i(P_i)u_i(t) \qquad (۲)$$

مسئلهٔ UC قیود مختلفی دارد که باید ارضا شوند [۲۶] از جمله:

- شرایط کارکرد اولیهٔ واحد تولیدی
- محدودیت‌های حداقل و حداکثر توان که به‌صورت زیر است:

$$P_{\min,i} \le P_i(t) \le P_{\max,i} \qquad (۳)$$

که در آن $P_{min,i}$، کمترین توان مجاز تولیدی توسط واحد iام و $P_{max,i}$بیشترین توان قابل تولید توسط واحد iام است.

- نرخ شیب افزایش یا کاهش توان

ثابت نرخ پاسخ واحد، تولید واحد را با محدودیت‌های تولید وابسته به زمان زیر محدود می‌کند:

$$P_{i,\max}(t) = \min\left\{P_{\max,i}, P_i(t-1) + \tau.RU_i\right\}$$
$$P_{i,\min}(t) = \max\left\{P_{\min,i}, P_i(t-1) - \tau.RD_i\right\} \qquad (۴)$$

که $P_{i,max}(t)$ حداکثر توان واحد iام در ساعت t، $P_{i,min}(t)$ حداقل توان واحد iام در ساعت t و $\tau = 60$ دقیقه در مراحل UC می‌باشد.

- قید تعادل توان در سیستم

در گام‌های یک دورهٔ برنامه‌ریزی مجموع توان‌های تولیدی توسط واحدها باید برابر با توان مورد نیاز سیستم باشد.

$$\sum_{i=1}^{N} u_i(t)P_i(t) = P_D(t) \qquad t = 1,2,...,H \qquad (۵)$$

- قید رزرو چرخان سیستم

برای حفظ پایداری سیستم در شرایط بحرانی، همواره

1. Straight Forward

مجموع توان‌های تولیدی باید بیشتر از توان مورد نیاز سیستم باشد. در اکثر مراجع، ۱۰ درصد از توان مورد نیاز به‌عنوان توان رزرو چرخان در سیستم در نظر گرفته می‌شود.

$$\sum_{i=1}^{N} u_i(t)\tilde{P}_{i,\max}(t) = P_D(t) + P_R(t) \qquad (۶)$$

که $\tilde{P}_{i,\max}(t)$ با استفاده از رابطهٔ (۴) به دست آمده و در آن τ برابر با ۱۰ دقیقه است. این ثابت نرخ پاسخ ده دقیقه‌ای برای توان اکتیو واحد تولیدی iام می‌باشد.

- حداقل زمان روشن یا خاموش بودن

$$\begin{aligned} T_i^c &\geq MUT_i \quad if \quad T_i^c \geq 0 \\ -T_i^c &\geq MDT_i \quad if \quad T_i^c \prec 0 \end{aligned} \qquad (۷)$$

که T_i^c معرف دوره‌های روشن و خاموش بودن واحد iام، MUT_i ماکزیمم زمان روشن بودن واحد i ام و MDT_i مینیمم زمان خاموش بودن واحد i ام هستند. جمع مقادیر T_i^c ها برای همهٔ واحدها باید برابر با افق برنامه‌ریزی باشد، به‌گونه‌ای که:

$$\sum_{c=1}^{C} \left| T_i^c \right| = H \qquad (۸)$$

۳. الگوریتم رقابت استعماری و اصلاح آن

الگوریتم های بهینه سازی مرسوم به‌طور عمده، الهام‌گرفته از فرایند های طبیعی هستند و در ارائهٔ آن‌ها به نمودهای تکامل انسانی توجهی نشده است. در الگوریتم رقابت استعماری، که اولین بار توسط لوکاس و آتش‌پز ارائه شده و برای بهینه‌سازی بعضی از توابع هزینه استفاده شده است [۲۱]، نه از یک پدیدهٔ طبیعی، بلکه از یک پدیدهٔ اجتماعی انسانی الهام گرفته شده است. به‌طور ویژه، این الگوریتم به فرایند استعمار، به‌عنوان مرحله‌ای از تکامل اجتماعی‌سیاسی بشر نگریسته و با مدل‌سازی ریاضی این پدیدهٔ تاریخی، از آن به‌عنوان منشأ الهام یک الگوریتم قدرتمند در زمینهٔ بهینه‌سازی بهره می‌گیرد. همانند دیگر الگوریتم‌های تکاملی، این الگوریتم نیز با تعدادی جمعیت اولیهٔ تصادفی که هرکدام از آن‌ها یک «کشور» نامیده می‌شوند، شروع می‌شود و سپس تعدادی از بهترین عناصر جمعیت به‌عنوان استعمارگر انتخاب شده و باقی‌ماندهٔ جمعیت نیز به‌عنوان مستعمره در نظر گرفته می‌شوند. استعمارگران بسته به قدرتشان، این مستعمرات را با یک روند خاص که در ادامه می‌آید، به‌سمت خود جذب می‌کنند. نقطهٔ پایانی رقابت استعماری، زمانی است که یک امپراطوری واحد با مستعمراتی که از لحاظ موقعیت، به خود کشور امپریالیست، خیلی نزدیک هستند، در دنیا داشته باشیم. مراحل این الگوریتم به‌صورت زیر است:

- شکل‌دهی امپراطوری‌های اولیه

در مسائل بهینه‌سازی، یافتن یک جواب بهینه برحسب متغیرهای مسئله، هدف نهایی است. در اولین گام، یک آرایه از متغیرهای مسئله که همان کشورها بوده و باید بهینه شوند تشکیل می‌شود. در یک مسئلهٔ بهینه‌سازی N_{VAR} بعدی، یک کشور، یک آرایهٔ $1*N_{VAR}$ است، که به‌صورت زیر تعریف می‌شود:

$$country = [P_1, P_2, ..., P_{N\,\mathrm{var}}]$$

هزینهٔ یک کشور با ارزیابی تابع f در متغیرهای P_1 تا P_{Nvar} یافته می‌شود که برابر است با:

$$\cos t_i = f\left(Country_i\right) = f(P_1, P_2, ..., P_{N\,\mathrm{var}}) \qquad (۹)$$

برای شروع الگوریتم، تعداد $N_{country}$ کشور اولیه ایجاد می‌شود. N_{imp} تا از بهترین اعضای این جمعیت، با کمترین مقدار تابع هزینه، به‌عنوان استعمارگر انتخاب می‌شوند. باقی‌ماندهٔ N_{col} تا از کشورها، مستعمراتی را تشکیل می‌دهند که هرکدام به یک امپراطوری تعلق دارند. تقسیم اولیهٔ مستعمرات، متناسب با قدرت استعمارگران است. برای انجام این کار، با داشتن هزینهٔ همهٔ استعمارگرها، هزینهٔ نرمالیزهٔ آن‌ها به‌صورت زیر در نظر گرفته می‌شود:

$$C_n = \max\{c_i\} - c_n \qquad (۱۰)$$

که در آن، c_n هزینهٔ استعمارگر nام، $max\{c_i\}$ بیشترین هزینه میان استعمارگرها و C_n هزینهٔ نرمالیزه‌شدهٔ این استعمارگر است. هر استعمارگری که دارای هزینه بیشتری باشد، هزینهٔ نرمالیزهٔ کمتری خواهد داشت. با داشتن هزینهٔ نرمالیزه، قدرت نسبی نرمالیزهٔ هر استعمارگر، به‌صورت رابطهٔ (۱۱) محاسبه شده و بر مبنای آن، کشورهای مستعمره بین امپریالسیت‌ها

تقسیم می‌شوند.

$$P_n = \left| \frac{C_n}{\sum_{i=1}^{Nimp} C_n} \right| \qquad (۱۱)$$

از دیدگاه دیگر، قدرت نرمالیزه‌شدهٔ یک امپریالیست، متناسب با مستعمراتی است که توسط آن امپریالیست اداره می‌شود. بنابراین تعداد اولیهٔ مستعمرات یک استعمارگر برابر خواهد بود با:

$$N.C_n = round\{P_n.N_{col}\} \qquad (۱۲)$$

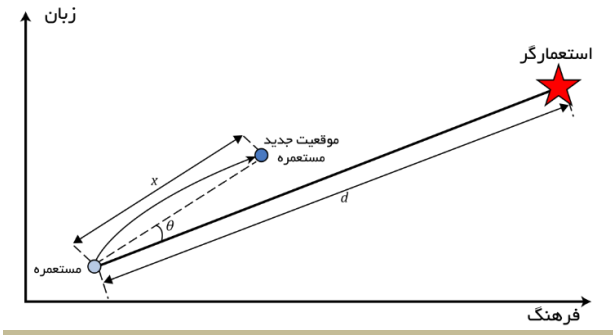
که در آن، $N.C_n$ تعداد اولیهٔ مستعمرات یک امپراطوری و N_{col} نیز تعداد کل کشورهای مستعمرهٔ موجود در جمعیت کشورهای اولیه است. تابع $round$ نیز تابعی است که خروجی آن نزدیک‌ترین عدد صحیح به عبارت داخل مجموعه است. برای هر امپراطوری، به تعداد $N.C_n$ کشور مستعمرهٔ اولیه را به‌صورت تصادفی انتخاب کرده و به استعمارگر شمارهٔ n می‌دهیم.

- حرکت مستعمره‌ها به‌سمت استعمارگر

این بخش از فرایند استعمار در الگوریتم بهینه‌سازی، به‌صورت حرکت مستعمرات به‌سمت کشور استعمارگر، مدل شده است. شکل (۱) شمای کلی این حرکت را نشان می‌دهد که مطابق آن، کشور استعمارگر، مستعمره‌ها را در راستای محورهای فرهنگ و زبان به‌سمت خود جذب می‌کند، به‌نحوی که کشور مستعمره به‌اندازهٔ x واحد در جهت خط واصل مستعمره به استعمارگر حرکت کرده و به موقعیت جدید مستعمره کشانده می‌شود. فاصلهٔ میان استعمارگر و مستعمره با d نشان داده شده است. x نیز عددی تصادفی با توزیع یکنواخت (و یا هر توزیع مناسب دیگر) است که برابر است با:

$$x = U(0, \beta * d) \qquad (۱۳)$$

و در آن β عددی بزرگ‌تر از ۱ و نزدیک به ۲ است. وجود ضریب $\beta>1$ باعث می‌شود تا کشور مستعمره در حین حرکت به‌سمت کشور استعمارگر از جهت‌های مختلف به آن نزدیک شود؛ بنابراین $\beta=2$ می‌تواند یک انتخاب مناسب باشد.



شکل (۱): شمای کلی حرکت مستعمرات به‌سمت استعمارگر [۲۱]

- قدرت کل یک امپراطوری

قدرت یک امپراطوری برابر است با قدرت کشور استعمارگر، به‌اضافهٔ درصدی از قدرت کل مستعمرات آن، بدین ترتیب هزینهٔ کل یک امپراطوری برابر است با:

$$TC_n = Cost(imperialist) + \xi\, mean\{Cost(colonies\ of\ empires)\} \qquad (۱۴)$$

که در آن، $T.C_n$ هزینهٔ کل امپراطوری nام و ξ عددی مثبت است که معمولاً بین صفر و یک و نیز مقداری نزدیک به صفر در نظر گرفته می‌شود.

- رقابت بین استعمارگران

همان گونه که پیش‌تر نیز بیان شد، هر امپراطوری که نتواند بر قدرت خود بیفزاید و قدرت رقابت خود را از دست بدهد، در جریان رقابت‌های استعمارگرانه، حذف خواهد شد. فرایند حذف شدن، به‌صورت تدریجی صورت می‌پذیرد و به مرور زمان، امپراطوری‌های ضعیف‌تر، مستعمرات خود را از دست داده و امپراطوری‌های قوی‌تر، این مستعمرات را تصاحب کرده و بر قدرت خویش می‌افزایند. شکل (۲) شمای کلی این بخش از الگوریتم را نشان می‌دهد.

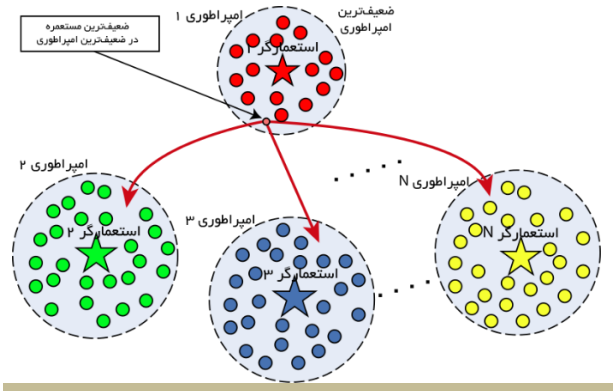
امپراطوری‌ها برای تصاحب این مستعمرات، ابتدا احتمال تصاحب هر امپراطوری را با در نظر گرفتن هزینهٔ کل امپراطوری محاسبه می‌کنند و سپس هزینهٔ نرمالیزه‌شدهٔ آن به‌صورت زیر محاسبه می‌شود:

$$N.T.C_n = \max\{T.C_i\} - T.C_n \qquad (۱۵)$$

که $T.C_n$ هزینهٔ کل امپراطوری nام و $N.T.C_n$ نیز هزینهٔ کل نرمالیزه‌شدهٔ آن امپراطوری است. مقدار $T.C_n$ معادل هزینهٔ کل یک امپراطوری و $N.T.C_n$ معادل قدرت کل آن است. به‌تعبیری دیگر،

امپراطوری با کمترین هزینه، دارای بیشترین قدرت است. با داشتن هزینهٔ کل نرمالیزه‌شده، احتمال تصاحب مستعمرهٔ رقابت، توسط هر امپراطوری، به‌صورت زیر محاسبه می‌شود:

$$P_{pu} = \left| \frac{N.T.C_n}{\sum_{i=1}^{Nimp} N.T.C_i} \right| \qquad (۱۶)$$



شکل (۲): شمای کلی تصاحب مستعمرات توسط امپراطوری‌های قوی‌تر [۲۱]

با داشتن احتمال تصاحب هر امپراطوری، برای اینکه مستعمرات مذکور را به‌صورت تصادفی، ولی با احتمال وابسته به قدرت هر امپراطوری بین آن‌ها تقسیم شود؛ بردار P از روی مقادیر احتمال فوق، به‌صورت زیرتشکیل می‌شود:

$$P = \left[P_{p1}, P_{p2}, ... P_{pNimp} \right] \qquad (۱۷)$$

بردار P دارای سایز $1*N_{imp}$ می‌باشد و از مقادیر احتمال تصاحب امپراطوری‌ها تشکیل شده است. سپس بردار تصادفی R هم‌سایز با بردار P تشکیل می‌شود. آرایه‌های این بردار، اعدادی تصادفی با توزیع یکنواخت در بازهٔ صفر و یک هستند.

$$r = \left[r_1, r_2, ..., r_{Nimp} \right]$$
$$r_1, r_2, ..., r_{Nimp} = U(0,1) \qquad (۱۸)$$

سپس بردار D را به‌صورت زیر تشکیل می‌شود:

$$D = P - R = \left[D_1, D_2, ..., D_{Nimp} \right] \Rightarrow$$
$$D = \left[P_{p1} - r_1, P_{p2} - r_2, ..., P_{pNimp} - r_{Nimp}, \right] \qquad (۱۹)$$

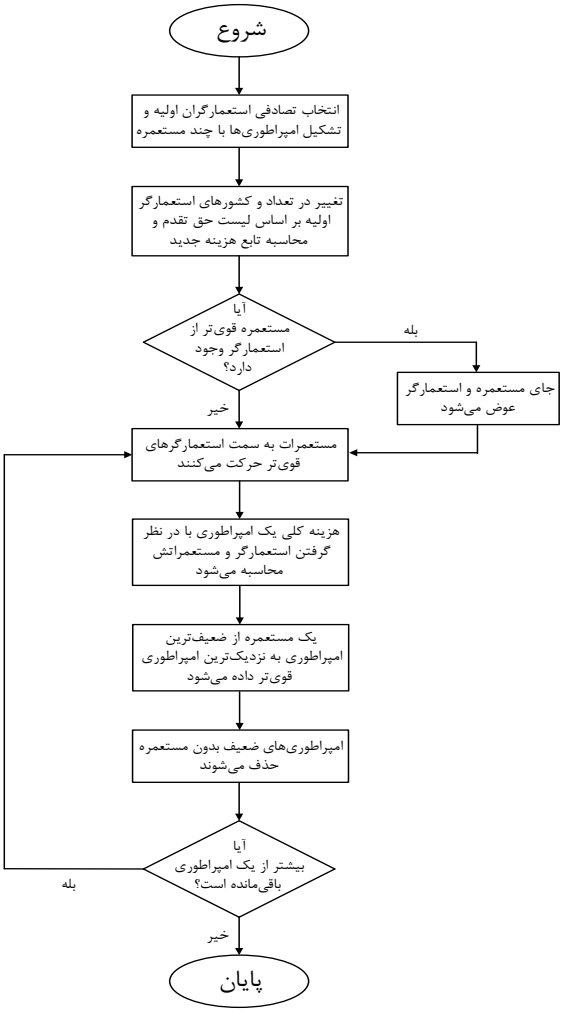
با داشتن بردار D مستعمرات مذکور به امپراطوری تعلق می‌گیرند که اندیس مربوط به آن در بردار D بزرگ‌تر از بقیه باشد. امپراطوری که بیشترین احتمال تصاحب را داشته باشد، اندیس مربوط به آن در بردار D بیشترین مقدار را خواهد داشت. با تصاحب مستعمره توسط یکی از امپراطوری‌ها، عملیات این مرحله از الگوریتم نیز به پایان می‌رسد.

۱.۳. الگوریتم رقابت استعماری اصلاح‌شده

الگوریتم رقابت استعماری اگرچه الگوریتمی بسیار قدرتمند برای بهینه‌سازی توابع چند متغیره است، ایراداتی مانند ضعف در جست‌وجوی کامل محیط در مسائل با ابعاد بزرگ و جابه‌جایی متعدد مستعمره‌ها بین استعمارگران در یک حلقهٔ بسته که اصطلاحاً به وضعیت قفل‌کردن الگوریتم مشهور می‌باشد، متوجه آن است. در الگوریتم رقابت استعماری، پس از تعیین تعداد مستعمره‌های هر امپراطوری طبق رابطهٔ (۱۲)، انتخاب مستعمرات به‌صورت تصادفی است، اما در این مقاله، تشکیل امپراطوری‌های اولیه با استفاده از لیست حق تقدم بوده به‌طوری که ابتدا متوسط هزینه به‌ازای ماکزیمم توان برای هر واحد را محاسبه کرده و هرکدام که کمتر باشند در اولویت قرار می‌گیرند. انتخاب استعمارگران اولیه طبق هزینهٔ نرمالیزه‌شدهٔ آن‌ها نبوده و تمام کشورها قابلیت استعمارگر بودن را دارند. ترکیب اولیهٔ استعمارگران به‌صورت تصادفی اتفاق افتاده و سپس طبق روش لیست حق تقدم، با جابه‌جایی‌های متفاوت، گروهی از کشورها انتخاب می‌شوند که در کنار مستعمرات اولیهٔ خود، بهترین تابع هزینه را داشته باشند. در این میان، حتی تعداد اولیهٔ امپراطوری‌ها نیز بعد از دور اول لیست‌های حق تقدم می‌تواند تغییر کند و مجموعه‌ای از کشورهای قوی‌تر به‌گونه‌ای انتخاب می‌شوند که به‌صورت جهت‌دار در راستای کمک به رقابت‌های استعماری بین امپراطوری‌ها بوده و این رقابت‌ها با سرعت و دقت بیشتری به نتایج مطلوب‌تر می‌رسند. فلوچارت این روش در شکل (۳) نشان داده شده است.

مقادیر پارامترهای بهینه در الگوریتم رقابت استعماری پس از استفاده از لیست حق تقدم برای سیستم تحت مطالعه به‌صورت زیر انتخاب شده‌اند:

$N_{imp} = 15, N_{col} = 150, \zeta = 0.2, \beta = 2$



شکل (۳): فلوچارت الگوریتم بهینه‌سازی رقابت استعماری اصلاح‌شده

۴. بررسی نتایج شبیه‌سازی

الگوریتم پیشنهادی (MICA) روی سیستم‌های شامل ۱۰، ۶۰ و ۱۰۰ واحدی IEEE برای تولید انرژی در افق برنامه‌ریزی ۲۴ ساعته مورد تست قرار گرفته است. ضرایب توابع هزینه و قیود واحدهای تولیدی برای سیستم مذکور در جدول (۱) و اطلاعات بار درخواستی از شبکه در مرجع [۲۶] آورده شده است، به‌طوری که برای سیستم ۱۰ واحدی اطلاعات بار در جدول (۲) آمده است.

حجم توان تولیدی تمامی واحدها به تفکیک ساعات مختلف شبانه‌روز در بخش پیوست (۱) ارائه شده و هزینه‌های راه‌اندازی و تولید آن‌ها محاسبه و در نهایت، هزینهٔ کل بهره‌برداری برای یک دورهٔ ۲۴ ساعته نشان داده می‌شود.

جدول (۱): ضرایب سیستم تحت مطالعهٔ IEEE [۲۶]

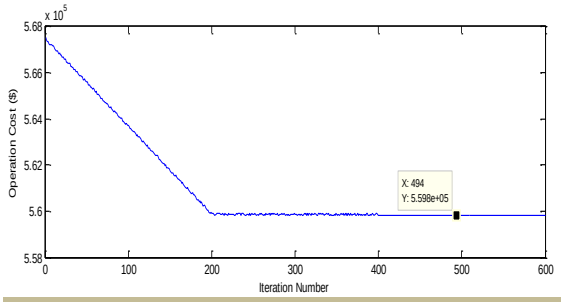
	واحد ۱	واحد ۲	واحد ۳	واحد ۴	واحد ۵
Pi_{max}	۴۵۵	۴۵۵	۱۳۰	۱۳۰	۱۶۲
Pi_{min}	۱۵۰	۱۵۰	۲۰	۲۰	۲۵
a_i	۱۰۰۰	۹۷۰	۷۰۰	۶۸۰	۴۵۰
b_i	۱۶/۱۹	۱۷/۲۶	۱۶/۶۰	۱۶/۵۰	۱۹/۷۰
c_i	۰/۰۰۰۴۸	۰/۰۰۰۳۱	۰/۰۰۲	۰/۰۰۲۱۱	۰/۰۰۳۹۸
MU_i	۸	۸	۵	۵	۶
MD_i	۸	۸	۵	۵	۶
$Hcost_i$	۴۵۰۰	۵۰۰۰	۵۵۰	۵۶۰	۹۰۰
$Ccost_i$	۹۰۰۰	۱۰۰۰۰	۱۱۰۰	۱۱۲۰	۱۸۰۰
$Chour_i$	۵	۵	۴	۴	۴
Inin state	۸	۸	-۵	-۵	-۶
	واحد ۶	واحد ۷	واحد ۸	واحد ۹	واحد ۱۰
Pi_{max}	۸۰	۸۵	۵۵	۵۵	۵۵
Pi_{min}	۲۰	۲۵	۱۰	۱۰	۱۰
a_i	۳۷۰	۴۸۰	۶۶۰	۶۶۵	۶۷۰
b_i	۲۲/۲۶	۲۷/۷۴	۲۵/۹۲	۲۷/۲۷	۲۷/۷۹
c_i	۰/۰۰۷۱۲	۰/۰۰۰۷۹	۰/۰۰۴۱۳	۰/۰۰۲۲۲	۰/۰۰۱۷۳
MU_i	۳	۳	۱	۱	۱
MD_i	۳	۳	۱	۱	۱
$Hcost_i$	۱۷۰	۲۶۰	۳۰	۳۰	۳۰
$Ccost_i$	۳۴۰	۵۲۰	۶۰	۶۰	۶۰
$Chour_i$	۲	۲	۰	۰	۰
Inin state	-۳	-۳	-۱	-۱	-۱

جدول (۲): توان مورد نیاز برای ۲۴ ساعت در سیستم شامل ۱۰ واحد [۲۶]

ساعت	بار	ساعت	بار
۱	۷۰۰	۱۳	۱۴۰۰
۲	۷۵۰	۱۴	۱۳۰۰
۳	۸۵۰	۱۵	۱۲۰۰
۴	۹۵۰	۱۶	۱۰۵۰
۵	۱۰۰۰	۱۷	۱۰۰۰
۶	۱۱۰۰	۱۸	۱۱۰۰
۷	۱۱۵۰	۱۹	۱۲۰۰
۸	۱۲۰۰	۲۰	۱۴۰۰
۹	۱۳۰۰	۲۱	۱۳۰۰
۱۰	۱۴۰۰	۲۲	۱۱۰۰
۱۱	۱۴۵۰	۲۳	۹۰۰
۱۲	۱۵۰۰	۲۴	۸۰۰

در تمام پخش بارهای صورت‌گرفته بین واحدها، قیود حداقل زمان روشن و حداکثر زمان خاموش بودن لحاظ شده و از ثابت جریمه‌ای برای این توابع استفاده نشده است. همان طور

که مشاهده می‌شود، هزینهٔ کل به‌دست‌آمده توسط روش پیشنهادی برابر با ۵۵۹۸۲۴ دلار بوده که در مقایسه با سایر روش‌های حل مسئله UC کمتر است. در بهترین نتیجهٔ به‌دست‌آمده توسط روش الگوریتم جست‌وجوی پرندهٔ فاخته، هزینهٔ بهره‌برداری برابر با ۵۶۱۲۸۰ دلار بوده که با اعمال تغییرات در الگوریتم و بهینه‌سازی آن، ۱۴۵۶ دلار صرفه‌جویی روزانه نتیجه می‌شود. مراجع [۲، ۸، ۱۳، ۱۷، ۱۹ و ۲۴ـ۲۶] به حل مسئلهٔ UC با استفاده از الگوریتم‌های هوشمند متفاوت پرداخته‌اند. هزینهٔ کلی بهره‌برداری و همچنین تفاوت آن با روش رقابت استعماری بهینه‌یافته در جدول (۳) مشاهده می‌شوند. نتایج نشان‌دهندهٔ عملکرد مطلوب‌تر این الگوریتم در مقایسه با تمامی روش‌های سابق است. در شکل (۴) مشخصهٔ همگرایی این الگوریتم آورده شده است. نتایج حاصل نشان می‌دهد که مقدار هزینهٔ بهینه به مقدار ۰/۷۳ درصد نسبت به سایر روش‌های مندرج در مرجع [۲۶] کمتر است؛ که نشان‌دهنده این است که به نقطهٔ بهینهٔ کلی نزدیک‌تر شده‌ایم. به‌طوری که این درصد در طول سال باعث صرفه‌جویی قابل توجهی می‌شود.

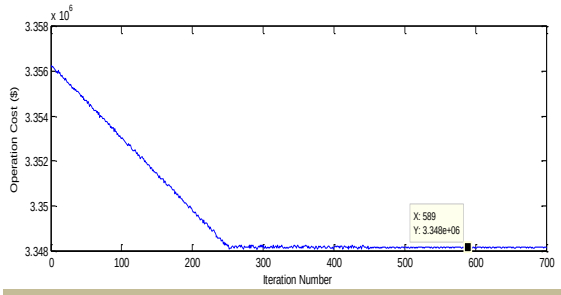


شکل (۴): مشخص همگرایی الگوریتم پیشنهادی در شبکه شامل ۱۰ واحد

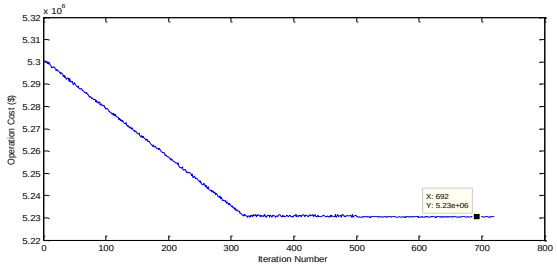
برای ارزیابی الگوریتم پیشنهادی، سیستم‌های تست با تعداد ۶۰ و ۱۰۰ واحد مطابق مرجع [۲۶] در نظر گرفته شده که نتایج حاصل به‌صورت مقایسه‌ای در جدول (۴) آورده شده است. همچنین در شکل‌های (۵) و (۶) روند نمودار مشخصهٔ همگرایی الگوریتم پیشنهادی آورده شده است. همان گونه که از نتایج مشهود است، استفاده از الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌های مندرج در مرجع [۲۶] به نقطهٔ بهینهٔ کلی نزدیک‌تر بوده و حداقل بهبودی توسط آن ۰/۷ درصد است.

جدول (۳): مقایسهٔ هزینه‌های تولید با استفاده از الگوریتم‌های مختلف

الگوریتم	اجتماع پرندگان بهینه‌یافته	سینوس- کسینوس	ژنتیک	اجتماع پرندگان دولایه
هزینهٔ نهایی (دلار)	۵۶۳۹۴۲	۵۶۳۹۳۷	۵۶۵۸۲۵	۵۶۳۷۰۵
صرفه‌جویی (دلار)	۴۱۱۸	۴۱۱۳	۶۰۰۱	۳۸۸۱
الگوریتم	جست‌وجوی پرندهٔ فاخته	ژنتیک بهبودیافته	کاوش باکتریایی	رقابت استعماری
هزینهٔ نهایی (دلار)	۵۶۱۲۸۰	۵۶۶۴۰۴	۵۶۴۸۴۲	۵۶۳۹۳۷
صرفه‌جویی (دلار)	۱۴۵۶	۶۵۰۸	۵۰۱۸	۴۱۱۳



شکل (۵): مشخصهٔ همگرایی الگوریتم پیشنهادی در شبکه شامل ۶۰ واحد



شکل (۶): مشخصهٔ همگرایی الگوریتم پیشنهادی در شبکه شامل ۱۰۰ واحد

جدول (۴): خلاصهٔ نتایج حاصل از الگوریتم پیشنهادی برای سیستم‌های تست شامل ۶۰ و ۱۰۰ واحد در مقایسه با مرجع [۲۶]

تعداد واحد	LR [26]	ICGA[26]	BF [26]	ICA [26]	MCIA
۶۰	۳۳۹۴۰۶۶	۳۳۷۸۱۰۸	۳۳۷۹۱۲۰	۳۳۷۱۷۲۲	۳۳۴۸۱۲۰
۱۰۰	۵۶۵۷۲۷۷	۵۶۳۰۸۳۸	۵۶۳۲۴۹۱	۵۶۱۷۹۱۳	۵۵۹۸۲۴

۵. نتیجه‌گیری

در این مقاله از الگوریتم رقابت استعماری اصلاح‌شده (MICA) برای حل مسئلهٔ UC شامل قیود مختلف از جمله حداقل زمان روشن و خاموش شدن، تعادل توان و رزرو چرخان ارائه شده است. کارایی روش ارائه‌شده با آزمایش بر روی سیستم تست IEEE ۱۰، ۶۰ و ۱۰۰ واحدی با

برنامه‌ریزی برای یک شبانه‌روز مورد بررسی قرار گرفته است. نتایج حاصل با نتایج بعضی از الگوریتم‌های هوشمند و روش‌های تحلیلی گذشته در حل مسئلهٔ UC مقایسه شده است؛ مقایسه‌ها پایین‌تر بودن هزینهٔ بهره‌برداری با این الگوریتم را نسبت به تمام روش‌های گذشته نشان می‌دهند.

اگرچه درصد اختلاف هزینه‌های بهره‌برداری در الگوریتم‌های مختلف هوشمند برای یک دورهٔ ۲۴ ساعته به نظر کم است، صرفه‌جویی‌های صورت‌گرفته در طول یک‌دوره زمانی گسترده مانند یک سال و یا بیشتر قابل توجه می‌باشد.

پیوست ۱

جدول تخصیص توان بین واحدها طی ۲۴ ساعت

St.Cost ($) راه اندازی	Load	Cost ($) بهره برداری	واحد تولید										ساعت
			۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	
۰	۷۰۰	۱۳۶۸۳	۰	۰	۰	۰	۰	۰	۰	۰	۲۴۵	۴۵۵	۱
۰	۷۵۰	۱۴۵۵۴	۰	۰	۰	۰	۰	۰	۰	۰	۲۹۵	۴۵۵	۲
۰	۸۵۰	۱۶۳۰۲	۰	۰	۰	۰	۰	۰	۰	۰	۳۹۵	۴۵۵	۳
۱۸۵۰	۹۵۰	۱۸۵۹۸	۰	۰	۰	۰	۰	۴۰	۰	۰	۴۵۵	۴۵۵	۴
۰	۱۰۰۰	۱۹۶۰۹	۰	۰	۰	۰	۰	۹۰	۰	۰	۴۵۵	۴۵۵	۵
۱۳۶۰	۱۱۰۰	۲۱۸۶۰	۰	۰	۰	۰	۰	۶۰	۱۳۰	۰	۴۵۵	۴۵۵	۶
۰	۱۱۵۰	۲۲۸۷۹	۰	۰	۰	۰	۰	۱۱۰	۱۳۰	۰	۴۵۵	۴۵۵	۷
۵۰۰	۱۲۰۰	۲۴۳۱۸	۰	۰	۰	۰	۲۰	۱۴۰	۱۳۰	۰	۴۵۵	۴۵۵	۸
۱۶۳۰	۱۳۰۰	۲۶۵۸۹	۰	۰	۰	۰	۲۰	۱۱۰	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۹
۰	۱۴۰۰	۲۸۷۶۸	۰	۰	۰	۰	۶۸	۱۶۲	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۱۰
۸۶۶	۱۴۵۰	۳۱۰۹۹	۰	۰	۱۳	۲۵	۸۰	۱۶۲	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۱۱
۹۰	۱۵۰۰	۳۳۰۸۵	۰	۱۰	۵۳	۲۵	۸۰	۱۶۲	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۱۲
۰	۱۴۰۰	۲۹۳۲۵	۰	۰	۰	۲۵	۴۳	۱۶۲	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۱۳
۰	۱۳۰۰	۲۶۵۸۹	۰	۰	۰	۰	۲۰	۱۱۰	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۱۴
۰	۱۲۰۰	۲۴۳۱۸	۰	۰	۰	۰	۲۰	۱۴۰	۱۳۰	۰	۴۵۵	۴۵۵	۱۵
۰	۱۰۵۰	۲۰۹۰۴	۰	۰	۰	۰	۰	۲۵	۱۱۵	۰	۴۵۵	۴۵۵	۱۶
۰	۱۰۰۰	۲۰۲۱۳	۰	۰	۰	۰	۰	۷۰	۲۰	۰	۴۵۵	۴۵۵	۱۷
۰	۱۱۰۰	۲۱۸۶۰	۰	۰	۰	۰	۰	۶۰	۱۳۰	۰	۴۵۵	۴۵۵	۱۸
۵۱۰	۱۲۰۰	۲۴۳۱۸	۰	۰	۰	۰	۲۰	۱۴۰	۱۳۰	۰	۴۵۵	۴۵۵	۱۹
۱۶۴۰	۱۴۰۰	۲۸۷۶۸	۰	۰	۰	۰	۶۸	۱۶۲	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۲۰
۰	۱۳۰۰	۲۶۵۸۹	۰	۰	۰	۰	۲۰	۱۱۰	۱۳۰	۱۳۰	۴۵۵	۴۵۵	۲۱
۰	۱۱۰۰	۲۲۴۹۱	۰	۰	۰	۰	۰	۴۰	۱۳۰	۲۰	۴۵۵	۴۵۵	۲۲
۰	۹۰۰	۱۷۸۶۰	۰	۰	۰	۰	۰	۰	۰	۲۰	۴۲۵	۴۵۵	۲۳
۰	۸۰۰	۱۶۱۱۱	۰	۰	۰	۰	۰	۰	۰	۲۰	۳۲۵	۴۵۵	۲۴
۸۴۴۶		**۵۵۱۳۷۸**							**هزینه کلی ۵۵۹۸۲۴ دلار**				

مراجع

[1] Pappala V. S. and Erlich I., "A new approach for solving the unit commitment problem by adaptive particle swarm optimization", IEEE Power and Energy Society General Meeting, pp. 1-6, 2008.

[2] Ting T. O., Rao M. V. C. and Loo C. K., "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization", *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 411-418, 2006.

[3] Yun-Won J., Jong-Bae P., Se-Hwan J. and Lee K. Y., "A New Quantum-Inspired Binary PSO for Thermal Unit Commitment Problems", 15th International Conference on Intelligent System Applications to Power Systems, pp. 1-6, 2009.

[4] Burns R., "Optimization of priority lists for a unit commitment program", in *Proc. IEEE Power Eng. Soc. Summer Meeting*, 1975.

[5] Sheble G. B., "Solution of the unit commitment problem by the method of unit periods", *IEEE Transactions on Power Systems*, vol. 5, no. 1, pp. 257-260, 1990.

[6] Snyder W. L., Powell H. D. and Rayburn J. C., "Dynamic programming approach to unit commitment", *IEEE Transactions on Power Systems*, vol. 2, no. 2, pp. 339-348, 1987.

[7] Ouyang Z. and Shahidehpour S. M., "An intelligent dynamic programming for unit commitment application", *IEEE Transactions on Power Systems*, vol. 6, no. 3, pp. 1203-1209, 1991.

[8] Bhadoria A., Marwaha S. and Kamboj V. K., "An optimum forceful generation scheduling and unit commitment of thermal power system using sine cosine algorithm", *Neural Computing and Applications*, vol. 32, no. 7, pp. 2785-2814, 2020.

[9] Zhuang F. and Galiana F. D., "Towards a more rigorous and practical unit commitment by Lagrangian relaxation", *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 763-773, 1988.

[10] Lee F. N., "A fuel-constrained unit commitment method", *IEEE Transactions on Power Systems*, vol. 4, no. 3, pp. 1208-1218, 1989.

[11] Virmani S., Adrian E. C., Imhof K. and Mukherjee S., "Implementation of a Lagrangian relaxation based unit commitment problem", *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1373-1380, 1989.

[12] Sheblé G. B. and Maifeld T. T., "Unit commitment by genetic algorithm and expert system", (in English), *Electric Power Systems Research*, Article vol. 30, no. 2, pp. 115-121, 1994.

[13] Kazarlis S. A., Bakirtzis A. G. and Petridis V., "A genetic algorithm solution to the unit commitment problem", *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 83-92, 1996.

[14] Senjyu T., Yamashiro H., Shimabukuro K., Uezato K. and Funabashi T., "Fast solution technique for large-scale unit commitment problem using genetic algorithm", *IEE Proceedings -- Generation, Transmission & Distribution*, vol. 150, no. 6, pp. 753-760, 2003.

[15] Zwe-Lee G., "Discrete particle swarm optimization algorithm for unit commitment", IEEE Power Engineering Society General Meeting, vol. 1, 2003.

[16] Zhai Y., Mu N., Liao X., Le J. and Huang T., "Unit commitment problem using an efficient pso based algorithm", in *2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, pp. 320-324, 2019.

[17] Zhai Y., Liao X., Mu N. and Le J., "A two-layer algorithm based on PSO for solving unit commitment problem", *Soft Computing*, 2019.

[18] Sisworahardjo N. S. and El-Keib A. A., "Unit commitment using the ant colony search algorithm", LESCOPE'02, 2002 Large Engineering Systems Conference on Power Engineering. Conference, pp. 2-6, 2002.

[19] Terki A. and Boubertakh H., "Application of a Binary-Real Coded Cuckoo Search Algorithm for Solving Unit Commitment Problem", in *2019 International Conference on Advanced Electrical Engineering (ICAEE)*, pp. 1-6, 2019.

[20] Hosseini S. H., Khodaei A. and Aminifar F., "A Novel Straightforward Unit Commitment Method for Large-Scale Power Systems", *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2134-2143, 2007.

[21] Atashpaz-Gargari E. and Lucas C., "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition", IEEE Congress on Evolutionary Computation, pp. 4661-4667, 2007.

[22] Biabangard-Oskouyi A., Atashpaz-Gargari E., Soltani N. and Lucas C., "Application of imperialist competitive algorithm for materials property characterization from sharp indentation test", *International Journal of Engineering Simulation*, vol. 10, no. 1, pp. 11-12, 2009.

[23] Khabbazi A., "Imperialist competitive algorithm for minimum bit error rate beamforming", International Journal of Bio-Inspired Computation, vol. 1, pp. 125-133, 2009.

[24] Damousis I. G., Bakirtzis A. G. and Dokopoulos P. S., "A solution to the unit-commitment problem using integer-coded genetic algorithm", *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1165-1172, 2004.

[25] Eslamian M., Hosseinian S. H. and Vahidi B., "Bacterial Foraging-Based Solution to the Unit-Commitment Problem", *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1478-1488, 2009.

[26] Hadji M. M. and Vahidi B., "A solution to the unit commitment problem using imperialistic competition algorithm", *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 117-124, 2011.