

به مدار آوردن نیروگاه‌ها با یک روش ابتکاری مبتنی بر الگوریتم تجمع زنبور عسل

ارسلان نجفی^۱، محسن فرشاد^{۲*}، حمید فلقی^۳

۱- دانشجوی کارشناسی ارشد، دانشکده فنی و مهندسی - دانشگاه بیرجند - بیرجند - ایران

۲- استادیار، دانشکده فنی و مهندسی - دانشگاه بیرجند - بیرجند - ایران

۳- استادیار، دانشکده فنی و مهندسی - دانشگاه بیرجند - بیرجند - ایران

چکیده

به مدار آوردن نیروگاه‌ها یکی از مسائل مهم در بهره‌برداری از سیستم‌های قدرت است که با توجه به محدودیت‌ها و پارامترهای زیاد مطرح در آن، از پیچیدگی بالایی برخوردار است. در این مقاله یک روش جدید مبتنی بر الگوریتم بهینه‌سازی تجمع زنبور عسل برای حل مسأله به مدار آوردن نیروگاه‌ها ارائه شده است که در آن از روش‌های کدبندی عدد صحیح (برای برآوری محدودیت‌های حداقل زمان روشن / خاموش) و کدبندی باینری (برای برآوری قید ذخیره چرخان) استفاده شده است. عدم استفاده از روش مرسوم ضرایب جریمه در برآوری قیود مسأله، از دیگر مزایای الگوریتم پیشنهادی است که از ترکیب روش‌های باینری و صحیح نتیجه شده است. با استفاده از الگوریتم پیشنهادی به خوبی می‌توان هزینه‌ی کل به مدار آوردن نیروگاه‌ها را کمینه کرد. نتایج شبیه‌سازی و مطالعات عددی گویای همگرایی بهتر الگوریتم پیشنهادی در مقایسه با سایر روش‌های موجود در این زمینه است.

واژه‌های کلیدی: الگوریتم تجمع زنبور عسل، بهره‌برداری از سیستم‌های قدرت، به مدار آوردن نیروگاه‌ها، بهینه‌سازی.

مقدمه

مسئله به مدار آوردن نیروگاه‌ها^۱ نقش مهمی در بهره‌برداری اقتصادی از سیستم‌های قدرت دارد. مشخص کردن زمان مناسب ورود یا خروج نیروگاه‌ها به مدار از بین حالت‌های ممکن، صرفه‌جویی عظیم اقتصادی در پی خواهد داشت. همچنین ایجاد ذخیره چرخان مناسب در سیستم قدرت، سیستم را از نظر امنیت در وضعیت مطلوبی نگه می‌دارد. مسئله UC در بر گیرنده‌ی برنامه روزانه کار نیروگاه‌هاست که در آن باید هزینه‌های بهره‌برداری و روشن/خاموش کردن نیروگاه‌ها کمینه شوند، در حالی که مجموعه محدودیت‌های حاکم رعایت شوند. این مسئله حاوی متغیرهای گسسته و پیوسته است که وضعیت روشن/خاموش بودن نیروگاه‌ها و نیز توزیع اقتصادی بار میان واحدها را نشان می‌دهد.

تا قبل از سال ۱۹۶۰ میلادی، مسئله UC فقط به بحث توزیع اقتصادی بار محدود می‌شد. در آن زمان، روش شناخته شده کوهن-تاکر وضعیت اقتصادی بهینه را مشخص می‌کرد. هنگامی که این شرایط برآورده می‌شد، کلیدی نیروگاه‌هایی که در مدار بودند، به غیر از نیروگاه‌هایی که به صورت مؤثر با حداکثر توان در شبکه بودند، با توجه به میزان سوختشان بارگذاری می‌شدند. بر مبنای روش کوهن-تاکر چندین راه حل برای مسئله UC پایه-گذاری گردید که از جمله آن‌ها می‌توان ابداع روش تکرار لامبدا و روش گرادیان برای نیروگاه‌های حرارتی را نام برد [۱].

در یک دسته‌بندی کلی، روش‌های موجود در حل مسئله UC را می‌توان به دو دسته‌ی تحلیلی و

هوشمند تقسیم کرد. از جمله روش‌های تحلیلی در حل مسئله UC می‌توان به روش معروف و قدیمی لاگرانژ اشاره کرد [۲]. در [۳-۲۶]، کاربرد روش‌های متنوع هوشمند در حل مسئله UC گزارش شده‌اند. در [۳ و ۴] روش‌هایی مبتنی بر کدبندی صحیح ارائه و از ضرایب جریمه برای برآوری بعضی از قیود استفاده شده است. در [۵ و ۶] از روش‌های کدبندی باینری برای حل مسئله UC استفاده شده است. همچنین می‌توان به کاربرد الگوریتم‌های تکاملی کوانتومی [۷]، باکتریال [۸]، ژنتیک [۹]، برنامه‌ریزی تکاملی [۱۰]، جهش قورباغه [۱۱]، سرد شدن تدریجی فلزات [۱۲]، تجمع مورچگان [۱۳] و دیگر روش‌های هوشمند [۱۴-۲۴] برای حل مسئله UC اشاره کرد. علاوه بر این در [۲۵] از ترکیب تکنیک‌های ریاضی و هوشمند برای حل مسئله UC استفاده شده است.

در [۲۶] روشی جدید مبتنی بر الگوریتم بهینه‌سازی تجمع زنبور عسل توسط نویسندگان برای حل مسئله UC ارائه شده است. در روش فوق از شیوه کدبندی صحیح برای به مدار آوردن نیروگاه‌ها استفاده شده است؛ ضمن آن که برآوری برخی از قیود مسئله با کاربرد روش مرسوم ضرایب جریمه محقق شده است.

در این مقاله روشی جدید مبتنی بر الگوریتم بهینه‌سازی تجمع زنبور عسل برای حل مسئله UC پیشنهاد شده است که در آن با بکارگیری شیوه‌ای نوین در کدبندی پاسخ‌های مسئله به صورت صحیح و باینری، قیود مسئله به صورت کارا و بی‌نیاز از کاربرد ضرایب جریمه رعایت شده است.

هزینه‌هایی هستند که در هنگام راه‌اندازی و خاموش کردن وجود دارند. هزینه خاموش کردن مقدار بسیار کمی است و عموماً در تحقیقات انجام شده از آن صرف‌نظر شده است. هزینه‌های راه‌اندازی در قالب ریاضی به صورت رابطه (۲) بیان می‌شود [۳]:

$$SU_i = \begin{cases} H_{\text{start up}} & T_i^{\text{OFF}} \leq MDT_i + (T_{\text{cold}}) \\ C_{\text{start up}} & T_i^{\text{ON}} > MDT_i + (T_{\text{cold}}) \end{cases} \quad (2)$$

که در آن، $H_{\text{start up}}$ هزینه راه‌اندازی گرم، $C_{\text{start up}}$ هزینه‌ی راه‌اندازی سرد، T_{cold} میزان زمانی که مشخص کننده صرفه اقتصادی واحد نیروگاهی برای راه‌اندازی گرم یا سرد است و همچنین MDT_i حداقل زمان مجاز فعالیت واحد نام است.

مجموعه قیودی که در این مسأله باید در نظر گرفته شود، به صورت زیر است:

- محدودیت حداقل زمان روشن و خاموش کردن که بیان کننده حداقل زمانی است که یک سیستم باید روشن یا خاموش باشد تا بتواند تغییر وضعیت بدهد:

$$\begin{cases} T_i^{\text{ON}} \geq MUT_i \\ T_i^{\text{OFF}} \geq MDT_i \end{cases} \quad \text{for } i = 1, 2, \dots, n \quad (3)$$

که در آن T_i^{ON} ، T_i^{OFF} و MUT_i به ترتیب بیانگر زمان روشن بودن، زمان خاموشی و حداقل زمان مجاز فعالیت واحد نام هستند.

- محدودیت تعادل بار که تضمین کننده تولید توان به میزان تقاضا است:

$$\sum_{i=1}^n u_i^t \cdot P_i^t = PD^t \quad \text{for } t = 1, 2, \dots, T \quad (4)$$

که در آن PD^t بیانگر میزان بار تقاضا در ساعت نام و T نیز کل دوره فعالیت است.

شبیه‌سازی‌های انجام گرفته بر روی چند سیستم نمونه گویای برتری روش پیشنهادی نسبت به دیگر روش‌های موجود در همگرایی به جواب بهینه است.

ساختار ادامه مقاله بدین شرح است: در بخش ۲، فرموله‌بندی مسأله UC ارائه شده است. در بخش ۳ الگوریتم بهینه‌سازی تجمع زنبور عسل تشریح و در بخش ۴ جزئیات نحوه بکارگیری آن در حل مسأله مورد نظر ارائه شده است. در بخش ۵ نتایج عددی حاصل از کاربرد روش پیشنهادی بر روی سه مثال نمونه با تعداد واحدهای مختلف و ذخیره چرخان‌های متفاوت ارائه و با دیگر روش‌ها مقایسه شده است و در نهایت مقاله با نتیجه‌گیری در بخش ۶ خاتمه می‌یابد.

فرمول‌بندی مسأله

هزینه بهره‌برداری در مسأله UC برابر با مجموع هزینه‌های سوخت، راه‌اندازی و خاموش کردن واحدهای نیروگاهی است که در یک بازه‌ی زمانی لازم است کمینه گردند. بنابراین، می‌توان گفت [۱۰]:

$$\min \sum_{t=1}^T \sum_{i=1}^n (FC_i(P_i^t) \cdot u_i^t) + SU_i + SD_i \quad (1)$$

در رابطه بالا FC_i هزینه سوخت واحد نیروگاهی نام، P_i^t توان تولیدی واحد i در ساعت نام، u_i^t وضعیت فعالیت یا عدم فعالیت واحد نیروگاهی نام در ساعت نام است که دارای دو حالت ۰ یا ۱ است، SU_i هزینه روشن کردن، SD_i هزینه خاموش کردن ژنراتور، T کل دوره برنامه‌ریزی و n تعداد واحدهای نیروگاهی را نشان می‌دهد. هزینه سوخت، قسمت اصلی مجموع هزینه‌ها را در بر می‌گیرد و هزینه‌های راه‌اندازی و خاموش کردن صرفاً

کاملاً به صورت تصادفی و بدون در نظر گرفتن تجربه بر می‌گزینند. هر منبع غذایی انتخاب شده بیانگر یک جواب ممکن در حل مسأله است. میزان شهد موجود در منابع غذایی بیانگر میزان برازندگی جواب مسأله است. تعداد زنبورهای کارگر مساوی با تعداد زنبورهای ناظر و برابر با تعداد جمعیت مسأله است. در این الگوریتم ابتدا جمعیت اولیه به صورت تصادفی و به تعداد NS تولید شده که NS بیانگر تعداد منابع غذایی و برابر با تعداد زنبورهای کارگر است. هر جواب $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ یک بردار n بعدی است. سپس این جمعیت وارد فرآیند جستجوی زنبورهای کارگر، ناظر و مأموران اکتشاف می‌شود.

در الگوریتم ABC تابع برازش به صورت زیر تعریف می‌شود:

$$fit_i = \begin{cases} \frac{1}{1 + f_i} & f \geq 0 \\ \frac{1}{1 + |f_i|} & f < 0 \end{cases} \quad (7)$$

که در آن f_i مقدار تابع هدف جواب i ام و fit_i برازندگی جواب i ام پس از تولید جواب‌های جدید است. زنبورهای ناظر منبع غذایی را با احتمال p_i انتخاب می‌کنند که در آن:

$$p_i = \frac{fit_i}{\sum_{j=1}^{NS} fit_j} \quad (8)$$

جواب‌های جدید به صورت زیر از جواب‌های قبلی تولید می‌شوند:

$$x_{ij}^{new} = x_{ij}^{old} + \lambda_{ij}(x_{ij}^{old} - x_{kj}^{old}) \quad \text{for } j=1,2,\dots,n \quad (9)$$

در رابطه فوق k عددی است تصادفی که از بازه $\{1,2,\dots,NS\}$ انتخاب می‌شود. λ_{ij} نیز عددی تصادفی در بازه $[-1, 1]$ است.

• محدودیت میزان تولید واحدهای تولیدی که به صورت زیر بیان می‌شود:

$$P_{i\min} \leq P_i \leq P_{i\max} \quad \text{for } i=1,2,\dots,n \quad (5)$$

که در آن $P_{i\min}$ و $P_{i\max}$ به ترتیب حداقل و حداکثر توان مجاز واحد تولیدی نام است.

• محدودیت ذخیره چرخان که بیانگر میزان ذخیره بالقوه واحدهای تولیدی است که هنوز به تولید نرسیده ولی می‌توانند حداکثر طی مدت ۱۰ دقیقه وارد مدار شوند. وجود ذخیره چرخان برای حفظ امنیت سیستم برای مقابله با برخی حوادث، از جمله خروج ناگهانی ژنراتورها و یا خطوط الزامی است. این قید را می‌توان به صورت ریاضی در قالب زیر بیان کرد:

$$\sum_{i=1}^n u_i^t \cdot P_{i\max}^t \geq PD^t + SR^t \quad (6)$$

که در آن SR^t میزان ذخیره چرخان در ساعت t ام است.

الگوریتم بهینه‌سازی تجمع زنبور عسل

الگوریتم تجمع زنبور عسل^۲ برای اولین بار در سال ۲۰۰۵ توسط Karaboga معرفی شد [۲۷]. این الگوریتم از شبیه‌سازی رفتار زنبورهای عسل در طبیعت به دست آمده و یکی از روش‌های بهینه‌سازی مبتنی بر جمعیت است. در این روش اجتماع زنبورها به سه گروه زنبورهای کارگر، ناظر و زنبورهای مأمور اکتشاف تقسیم می‌شود. زنبورهای کارگر به صورت تصادفی به دنبال منابع غذایی می‌گردند و خود را به اشتراک می‌گذارند. در این میان، زنبورهای ناظر از بین این منابع غذایی، با توجه به تجربه و موقعیت خود، منبع غذایی مناسب را انتخاب می‌کنند، در حالی که زنبورهای مأمور اکتشاف، منابع غذایی را

کدبندی باینری و صحیح است. در این روش تعداد سیکل‌های روشن / خاموش واحدها ۲۴ در نظر گرفته شده است چراکه با محدود کردن تعداد سیکل‌ها به عدد ۵ (که در کارهای قبلی مرسوم بود) نمی‌توان در طی برنامه به صورت آزادانه واحدها را روشن و یا خاموش کرد [۲۶]. براساس این روش تولید جمعیت اولیه، تولید جواب‌های جدید و برآورده کردن حداقل زمان روشن / خاموش به صورت صحیح و برآورده شدن محدودیت ذخیره‌ی چرخان و خارج کردن ظرفیت‌های اضافه به صورت باینری کد می‌شوند. فلوجارت روش پیشنهادی در شکل (۱) نشان داده شده است.

تولید جمعیت اولیه

در این مقاله، نحوه‌ی تولید جمعیت اولیه و سپس ترکیب آن‌ها به گونه‌ای است که کاربرد فاکتور جریمه در خصوص عدم برآوری قیود حداقل زمان روشن / خاموش را از بین می‌برد. طی این فرآیند، جمعیت اولیه به صورت اعدادی صحیح تولید می‌شوند. برای نشان دادن میزان ساعت روشن یا خاموش به ترتیب از اعداد صحیح مثبت و منفی استفاده شده است، همچنین عدد صفر به منزله‌ی تمام شدن دوره فعالیت واحد در ۲۴ ساعت است [۱۱]. با توجه به اینکه تعداد سیکل‌ها حداکثر برابر با تعداد ساعت‌های برنامه‌ریزی (در اینجا ۲۴) در نظر گرفته شده است، اعداد مثبت و منفی تولید شده به میزانی است که مجموع قدرمطلق آنها ۲۴ شود و قاعدتاً مابقی سیکل‌ها صفر فرض خواهند شد. در شکل (۲) یک نمونه جواب از جمعیت اولیه، شامل وضعیت

موقعیت‌های جدید، پس از تولید و برآزش، با موقعیت‌های قدیم مقایسه شده و اگر دارای کیفیت بهتری (شهد بیشتر) باشند، جایگزین آنها خواهند شد. ضمناً اگر موقعیتی بهبود پیدا نکرد، آن منبع غذایی متروکه اعلام و با منبع جدیدی که توسط زنبورهای اکتشاف مطابق رابطه‌ی زیر معرفی می‌شود، جایگزین خواهد شد:

$$x_{ij}^{new} = l_j + rand \cdot (u_j - l_j) \quad (10)$$

که در آن l_j و u_j حد پایین و بالای متغیر x_{ij} و $rand$ نیز عددی تصادفی بین صفر و یک هستند [۲۹، ۲۷].

پس از این که جمعیت جدید تولید و برآزش شدند، از بین جمعیت جدید عمل انتخاب صورت می‌گیرد. این اعمال تا زمانی که تعداد تکرارهای الگوریتم به پایان برسد، ادامه خواهد داشت.

حل مسأله UC مبتنی بر الگوریتم تجمع زنبور عسل

به مدار آوردن نیروگاه‌ها مسأله‌ای غیر خطی است و به دلیل دارا بودن قیود زیاد از پیچیدگی بسیار بالایی برخوردار است. به همین دلیل، روش‌های معمول با مشکلات زیادی در حل این مسأله روبه‌رو هستند و عموماً یا قادر به حل این مسأله نیستند و یا با سختی‌های زیادی در حل آن مواجه هستند. بنابر دلایل ذکر شده در این مقاله از الگوریتم بهینه‌سازی تجمع زنبور عسل استفاده شده است که برای حل مسائل غیر خطی بسیار کارا است. همچنین در این مقاله روشی جدید برای حل مسأله UC در نظر گرفته شده است، که ترکیبی از روش‌های

روشن / خاموش N واحد نیروگاهی طی یک دوره زمانی ۲۴ ساعته نشان داده شده است؛ ضمن اینکه برای شفافیت بیشتر موضوع، جزئیات کامل سیکل زمانی روشن / خاموش واحد ۱ در قسمت پایین شکل به صورت باز شده به تصویر کشیده شده است. برتری این کد نسبت به روش‌های تولید باینری در این است که خیلی از حالت‌های ناممکن در تولید جمعیت اولیه حذف می‌شوند. بر همین اساس، روش زیر برای تولید جمعیت اولیه معرفی می‌شود. برای تولید مقدار سیکل اول از روش زیر استفاده می‌شود:

$$T_i^1 = \begin{cases} rand((MUT_i - T_i^0), T) & \text{if } T_i^0 > 0 \\ -rand((MDT_i - T_i^0), T) & \text{if } T_i^0 < 0 \end{cases} \quad (11)$$

که در آن T_i^0 شرایط اولیه واحد نام است. برای سیکل‌های بعدی به این صورت عمل می‌شود:

شروع

تولید جمعیت اولیه

توزیع اقتصادی بار و ارزیابی برازندگی

تولید جمعیت جدید

برآوری قیود:
- ذخیره چرخان
- حداقل زمان روشنایی / خاموشی

خارج کردن ظرفیت‌های اضافه

توزیع اقتصادی بار و ارزیابی برازندگی

انتخاب

شرایط همگرایی برآورده شده است؟

خیر

بله

انتخاب جواب بهینه

پایان

شکل (۱): فلوچارت روش پیشنهادی

Unit 1					Unit 2					...	Unit N				
T_1^1	T_1^2	...	T_1^{23}	T_1^{24}	T_2^1	T_2^2	...	T_2^{23}	T_2^{24}	...	T_N^1	T_N^2	...	T_N^{23}	T_N^{24}
5	-3	...	0	0	24	0	...	0	0	...	8	-6	...	0	0

T_1^1	T_1^2	T_1^3	T_1^4	T_1^5	T_1^6	T_1^7	T_1^8	T_1^9	T_1^{10}	T_1^{11}	T_1^{12}	T_1^{13}	T_1^{14}	T_1^{15}	T_1^{16}	T_1^{17}	T_1^{18}	T_1^{19}	T_1^{20}	T_1^{21}	T_1^{22}	T_1^{23}	T_1^{24}
5	-3	3	-4	2	-3	3	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

شکل (۲): یک نمونه جواب از جمعیت اولیه شامل وضعیت روشن / خاموش N واحد نیروگاهی طی یک دوره زمانی

۲-۴- برآوری قید ذخیره چرخان

در حل باینری برای هر واحد نیروگاهی طی شبانه‌روز ۲۴ عدد باینری صفر یا یک وجود دارد که عدد صفر نشان دهنده خارج از مدار بودن و عدد یک بیانگر در مدار بودن واحد است. بر این اساس و برای ساعت‌هایی که در آن‌ها میزان ذخیره چرخان رعایت نشده باشد، واحدهای خاموش طبق فهرست حق تقدم (بر حسب تولید ارزان‌تر)، وارد مدار می‌شوند تا زمانی که میزان ذخیره چرخان در هر ساعت برآورده شود [۲۳].

برآوری قیود حداقل زمان روشن / خاموش

پس از تولید جمعیت باید قیدهای حداقل زمان روشن / خاموش در همه سیکل‌های زمانی بررسی شوند و در صورتی که این قیود نقض شده باشند ترمیم گردند. طبق شکل (۳) برای هر واحد نیروگاهی محدوده‌های مجاز و ممنوعه‌ای وجود دارد که روش پیشنهادی حالات غیر مجاز را به حالات مجاز تبدیل می‌کند. این کار به روش زیر انجام می‌شود [۴]:

برای برآوری حداقل زمان روشن اگر $T_i^c < MUT_i$ آنگاه داریم:

$$T_i^c = T_i^c + MUT_i \quad (15)$$

به صورت مشابهی برای برآوری حداقل زمان خاموش اگر $T_i^c < MDT_i$ آنگاه:

$$T_i^c = T_i^c - MDT_i \quad (16)$$

بدیهی است که در سیکل آخر اگر زمان باقیمانده کمتر از حداقل زمان روشنایی یا خاموشی واحد شود آنگاه رابطه زیر را خواهیم داشت:

اگر $T_i^{c-1} < 0$ باشد، سیکل بعدی یعنی T_i^c روشن می‌شود:

$$T_i^c = \begin{cases} rand(MUT_i, RT_i^{c-1}) & RT_i^{c-1} > MUT_i \\ RT_i^{c-1} & otherwise \end{cases} \quad (12)$$

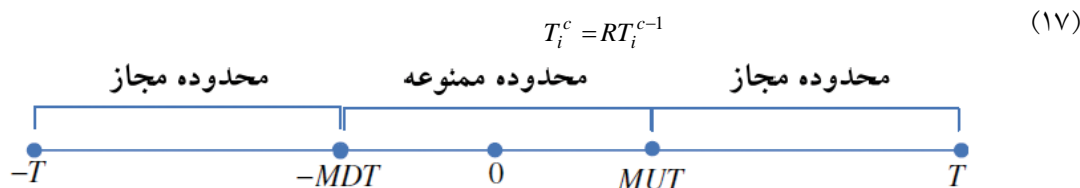
و اگر $T_i^{c-1} > 0$ باشد، سیکل بعدی یعنی T_i^c خاموش می‌شود:

$$T_i^c = \begin{cases} -rand(MDT_i, RT_i^{c-1}) & RT_i^{c-1} > MDT_i \\ -RT_i^{c-1} & otherwise \end{cases} \quad (13)$$

در روابط بالا RT_i^{c-1} طبق رابطه زیر به دست می‌آید:

$$RT_i^{c-1} = T - \sum_{k=1}^{c-1} |T_i^k| \quad (14)$$

همان طور که قبلاً ذکر شد، مجموع قدرمطلق سیکل‌ها باید برابر با تعداد ساعت‌های برنامه‌ریزی (مثلاً ۲۴ ساعت) باشد. بعد از تولید جمعیت‌های جدید این احتمال وجود دارد که مجموع حالت‌های روشن / خاموش متفاوت از ۲۴ شود. به همین دلیل، مقدار اضافه‌تر یا کمتر از ۲۴ به صورت نسبی بین جواب‌های جدید اضافه یا از آن‌ها کم خواهد شد. تابع $rand$ اعداد تصادفی بین صفر و یک تولید می‌کند. در نتیجه جمعیت جدید تولید شده صحیح نیستند، در حالی که در مسأله UC باید اعداد صحیح باشند. به همین دلیل، جمعیت تولید شده باید گرد شود. پس از گرد شدن جواب‌ها مجدداً این احتمال وجود دارد که مجموع حالت‌های خاموش یا روشن بیشتر از ۲۴ یا کمتر شود که در این صورت این اختلاف را در آخرین سیکل غیر صفر تأثیر داده تا مجموع کماکان برابر زمان برنامه‌ریزی (۲۴ ساعت) باشد [۱۱].



شکل (۳): موقعیت محدوده‌های مجاز و ممنوعه برای حداقل زمان روشنایی و خاموشی [۴]

که در آن a_i ، b_i و c_i ضرایب هزینه سوخت واحد نام هستند.

مطالعات عددی و نتایج

در این قسمت کارایی الگوریتم پیشنهادی در حل مسأله‌ی UC، با ارائه سه مثال نمونه و سپس مقایسه نتایج با سایر الگوریتم‌های موجود نشان داده شده است. شبیه‌سازی‌ها در کامپیوتری پنتیوم ۴ با پردازشگر ۲ گیگاهرتز و حافظه ۳ مگابایت انجام گرفته است.

(مثال ۱)

این مثال دارای ۱۰ واحد نیروگاهی است و از مرجع [۱۱] گرفته شده است. مشخصات این واحدها و میزان تقاضای بار ۲۴ ساعته در قسمت ضمیمه آمده است. میزان ذخیره‌ی چرخان در این مثال ۱۰٪ از بار هر ساعت در نظر گرفته شده است.

نتایج ۵۰ اجرای متوالی الگوریتم پیشنهادی بازای جمعیت‌های مختلف در جدول (۱) و نتایج مقایسه‌ای آن با سایر روش‌ها در جدول (۲) آورده شده است. نتایج جدول (۱) نشان می‌دهد که روش پیشنهادی با جمعیت اولیه‌ی ۳۰، ۵۰ و ۷۰ همواره به یک جواب دست یافته است که این مهم بیانگر کارایی الگوریتم پیشنهادی در حل مسأله‌ی UC است. همچنین با جمعیت اولیه ۱۰ به جواب بهینه

از بین بردن ظرفیت اضافه

همان طور که در بخش ۴-۲ ذکر شد، در روش پیشنهادی قید ذخیره‌ی چرخان با وارد کردن واحدهای جدید به مدار برآورده می‌شود. به همین دلیل، امکان به وجود آمدن ظرفیت اضافه (و لذا افزایش هزینه تولید) در سیستم وجود دارد. برای از بین بردن ظرفیت‌های اضافه در هر ساعت، طبق فهرست حق تقدم، واحدهای گرانتر از مدار خارج خواهند شد. این کار تا جایی ادامه می‌یابد که واحدهای فعال بتوانند بار و ذخیره‌ی چرخان را تأمین کرده و ضمناً قیود روشن / خاموش نقض نشوند [۲۳].

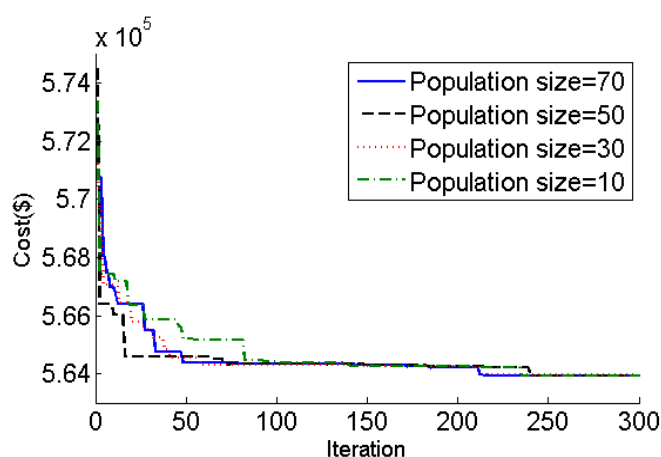
توزیع اقتصادی بار

پس از خارج کردن ظرفیت‌های اضافه از مدار، طبق روش کلاسیک تکرار n [۳۰]، توزیع اقتصادی بار انجام می‌گیرد. در این مرحله میزان توان تولیدی برای هر واحد نیروگاهی در مدار در هر ساعت مشخص می‌شود. میزان توان تولید شده از طریق توزیع اقتصادی بار، میزان هزینه تولید را طبق رابطه زیر مشخص خواهد کرد:

$$FC_i(P_i^t) = a_i(P_i^t)^2 + b_i P_i^t + c_i \quad (18)$$

اطلاعات مندرج در جدول (۲) نشان می‌دهند که این الگوریتم به جواب ۵۶۳۹۳۷/۶۴ دست یافته، در حالی که هیچ کدام از الگوریتم‌های ذکر شده به این جواب نرسیده‌اند. افزون بر این، الگوریتم پیشنهادی در مقایسه با روش‌های صرفاً باینری، Discrete Improved binary differential evolutionary و binary particle swarm optimization و روش‌های کدبندی با اعداد صحیح، Integer coded genetic algorithm، به جواب بهتری دست یافته است.

مورد نظر دست یافته ولی در تمامی اجراهای برنامه با این جمعیت اولیه، به جواب بهینه همگرا نشده است در حالی که با افزایش تعداد جمعیت همواره به یک عدد همگرا شده است. نمودار همگرایی الگوریتم پیشنهادی با جمعیت‌های مختلف در شکل (۴) نشان داده شده است. نتایج کامل اجرای روش پیشنهادی، شامل: برنامه ۲۴ ساعته فعالیت واحدها، توزیع اقتصادی توان، میزان ذخیره چرخان در هر ساعت و بالاخره هزینه‌های راه‌اندازی و بهره‌برداری هر ساعت در جدول (۳) آمده است.



شکل (۴): نمودار همگرایی با جمعیت‌های مختلف در مثال ۱

جدول (۱): نتایج اجرای روش پیشنهادی با جمعیت‌های مختلف در مثال ۱

جمعیت اولیه	بهترین جواب (\$)	میانگین جواب‌ها (\$)	بدترین جواب (\$)	زمان (ثانیه)
10	563937.64	564369.31	567310.76	43.72
30	563937.64	563937.64	563937.64	133.74
50	563937.64	563937.64	563937.64	221.25
70	563937.64	563937.64	563937.64	310.00

جدول (۲): مقایسه‌ی نتایج الگوریتم پیشنهادی با سایر روش‌ها در مثال ۱

روش	بهترین جواب (\$)
Lagrange relaxation [۲]	565825
Integer coded genetic algorithm [۳]	566604
Simulated annealing [۱۲]	565824
Genetic algorithm [۹]	565825
Ant colony search algorithm [۱۳]	564049
Discrete binary differential evolutionary [۶]	563977
Improved quantum evolutionary algorithm [۲۳]	563977
Improved binary particle swarm optimization[۵]	563977
Evolutionary programming [۱۰]	563977
Extended priority list [۱۸]	563977
Quantum evolutionary algorithm [۷]	563938
Artificial bee colony	563938

مثال ۲)

باعث افزایش تعداد حالات ممکن برای جواب مسأله می‌شود و در واقع، دلیل همگرا نشدن الگوریتم همواره به یک عدد در این مثال، افزایش تعداد حالات ممکن نسبت به مثال قبل (۱۰٪ ذخیره چرخان) است. همچنین نمودار همگرایی این الگوریتم با تعداد زنبورهای ۱۰، ۳۰، ۵۰ و ۷۰ در شکل (۵) موجود است. نتایج مقایسه‌ای مندرج در جدول (۵) گویای این واقعیت است که الگوریتم پیشنهادی توانسته است به جواب بهتری نسبت به الگوریتم‌های Adaptive Genetic algorithm، Two-stage particle swarm optimization، Improved particle genetic based technique، Three stage modification swarm optimization، process دست یابد. افزون بر این، مشاهده می‌شود

در این مثال، همان ۱۰ واحد نیروگاهی مثال ۱، البته با ملاحظه ۵٪ ذخیره چرخان مورد توجه قرار گرفته است. نتایج ۵۰ اجرای متوالی الگوریتم پیشنهادی با جمعیت‌های مختلف در جدول (۴) نشان داده شده است. نتایج مندرج در این جدول نشان می‌دهد که الگوریتم پیشنهادی حتی با تعداد جمعیت اولیه کم نیز به جواب بهینه دست یافته است ولی با افزایش تعداد جمعیت تعداد دفعات رسیدن به جواب بهینه افزایش پیدا کرده است که نهایتاً این افزایش جمعیت موجب بهبود میانگین اجراهای مختلف شده است. علاوه بر این، از مقایسه نتایج این جدول با جدول (۱) می‌توان به این نتیجه رسید که در نظر گرفتن ۵٪ ذخیره‌ی چرخان به جای ۱۰٪،

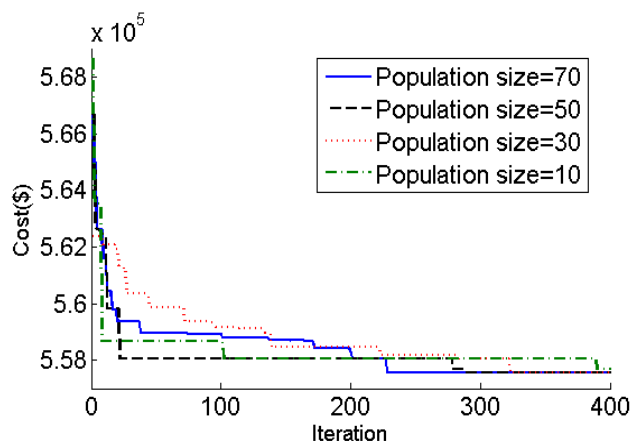
که برخی از این روش‌ها به جواب‌هایی بسیار دورتر از جواب بهینه همگرا شده‌اند. نتایج کامل اجرای روش پیشنهادی، شامل: برنامه ۲۴ ساعته فعالیت واحدها، توزیع اقتصادی توان، میزان ذخیره چرخان در هر ساعت و بالاخره هزینه‌های راه‌اندازی و بهره‌برداری هر ساعت در جدول (۶) آمده است.

جدول (۳): نحوه توزیع توان در ۲۴ ساعت در مثال ۱ (۱۰٪ ذخیره چرخان)

ذخیره‌ی چرخان (%)	هزینه بهره‌برداری (\$) (\$)	هزینه راه‌اندازی (\$) (\$)	واحد										ردیف	
			10	9	8	7	6	5	4	3	2	1		
30	13683.12	0	0	0	0	0	0	0	0	0	0	245	455	1
21.33	14554.5	0	0	0	0	0	0	0	0	0	0	295	455	2
26.12	16809.45	900	0	0	0	0	0	25	0	0	0	370	455	3
13.55	18597.68	0	0	0	0	0	0	40	0	0	0	455	455	4
20.2	20020.02	560	0	0	0	0	0	25	130	0	0	390	455	5
21.09	22387.04	1100	0	0	0	0	0	25	130	130	0	360	455	6
15.82	23261.99	0	0	0	0	0	0	25	130	130	0	410	455	7
11	24150.32	0	0	0	0	0	0	30	130	130	0	455	455	8
15.15	27251.07	860	0	0	0	25	20	85	130	130	0	455	455	9
10.85	30057.54	60	0	0	10	25	33	162	130	130	0	455	455	10
10.83	31916.07	60	0	10	10	25	73	162	130	130	0	455	455	11
11.47	33890.15	60	10	10	43	25	80	162	130	130	0	455	455	12
10.85	30057.54	0	0	0	10	25	33	162	130	130	0	455	455	13
15.15	27251.07	0	0	0	0	25	20	85	130	130	0	455	455	14
11	24150.32	0	0	0	0	0	0	30	130	130	0	455	455	15
26.85	21513.64	0	0	0	0	0	0	25	130	130	0	310	455	16
33.2	20641.83	0	0	0	0	0	0	25	130	130	0	260	455	17
21.09	30057.54	0	0	0	0	0	0	25	130	130	0	360	455	18
11	27251.07	0	0	0	0	0	0	30	130	130	0	455	455	19
10.86	30057.54	490	0	0	10	25	33	162	130	130	0	455	455	20
15.15	27251.07	0	0	0	0	25	20	85	130	130	0	455	455	21
12.45	22735.54	0	0	0	0	25	20	145	0	0	0	455	455	22
10	17645.36	0	0	0	0	0	20	0	0	0	0	425	455	23
13.75	15427.43	0	0	0	0	0	0	0	0	0	0	345	455	24
559847.6		4090	هزینه کل: 563937.644											

ژنراتور است، انجام و نتایج آن گزارش شده است. این مثال با تکرار اطلاعات واحدهای مثال ۱ و با دو برابر کردن میزان بار تقاضا به دست آمده است.

مثال ۳) برای نشان دادن توانایی روش پیشنهادی، اجرای آن بر روی یک مثال با ابعاد بزرگتر که دارای ۲۰



شکل (۵): نمودار همگرایی با جمعیت‌های مختلف در مثال ۲

در حالی که هیچ کدام از روش‌های ذکر شده دیگر موفق به دستیابی به پاسخ مزبور نبوده است. این امر برتری و کارایی روش پیشنهادی را در حل مسأله نشان می‌دهد.

نتایج ۵۰ اجرای متوالی الگوریتم پیشنهادی با جمعیت‌های مختلف در جدول (۷) و نتایج مقایسه‌ای آن با سایر روش‌ها در جدول (۸) آورده شده است. نتایج ذکر شده نشان می‌دهند که الگوریتم پیشنهادی تونسته است به جواب بهینه ۱۱۲۴۲۹۳/۵۴ دست یابد،

جدول (۴): نتایج اجرای روش پیشنهادی با جمعیت‌های مختلف در مثال ۲

جمعیت اولیه	بهترین جواب (\$)	میانگین جواب‌ها (\$)	بدترین جواب (\$)	زمان (ثانیه)
10	557563.41	558081.60	558681.06	50.82
30	557563.41	557887.07	558358.37	120.25
50	557563.41	557687.98	557991.83	199.50
70	557563.41	557658.82	557991.83	253.50

جدول (۵): مقایسه نتایج الگوریتم پیشنهادی با سایر روش‌ها در مثال ۲

روش	بهترین جواب (\$)
Genetic algorithm [۹]	570871
Adaptive particle swarm optimization [۲۰]	561586
Two-stage genetic based technique [۲۱]	564450
Improved particle swarm optimization [۲۲]	558114
Three stage modification process [۲۴]	557676.81
Artificial bee colony	557563.41

جدول (۶): نحوه توزیع توان در ۲۴ ساعت در مثال ۲ (۵٪ ذخیره چرخان)

ذخیره چرخان (%)	هزینه بهره‌برداری (\$)	هزینه راه‌اندازی (\$)	واحد										ردیف	
			10	9	8	7	6	5	4	3	2	1		
30	13683.12	0	0	0	0	0	0	0	0	0	0	245	455	1
21.33	14554.5	0	0	0	0	0	0	0	0	0	0	295	455	2
7.05	16301.88	0	0	0	0	0	0	0	0	0	0	395	455	3
9.47	18637.69	560	0	0	0	0	0	0	130	0	0	365	455	4
20.2	20020.02	900	0	0	0	0	0	25	130	0	0	390	455	5
9.27	21860.27	0	0	0	0	0	0	60	130	0	0	455	455	6
15.82	23261.99	1100	0	0	0	0	0	25	130	130	0	410	455	7
11	24150.32	0	0	0	0	0	0	30	130	130	0	455	455	8
8.61	26588.96	340	0	0	0	0	20	110	130	130	0	455	455	9
6.93	29365.94	520	0	0	0	25	43	162	130	130	0	455	455	10
7.03	31219.61	60	0	0	13	25	80	162	130	130	0	455	455	11
7.13	33205.27	60	0	10	53	25	80	162	130	130	0	455	455	12
6.93	29365.94	0	0	0	0	25	43	162	130	130	0	455	455	13
5.15	27166.73	0	0	0	0	25	73	162	130	0	0	455	455	14
6.83	24318.01	0	0	0	0	0	20	140	130	0	0	455	455	15
14.47	20895.9	0	0	0	0	0	0	25	130	0	0	440	455	16
20.2	20020.02	0	0	0	0	0	0	25	130	0	0	390	455	17
9.27	21860.27	0	0	0	0	0	0	60	130	0	0	455	455	18
11	24150.32	550	0	0	0	0	0	30	130	130	0	455	455	19
6.93	29365.94	430	0	0	0	25	43	162	130	130	0	455	455	20
15.15	27251.07	0	0	0	0	25	20	85	130	130	0	455	455	21
9.54	22576.93	0	0	0	0	25	35	0	0	130	0	455	455	22
15.55	17795.28	0	0	0	0	0	0	0	0	130	0	315	455	23
13.75	15427.43	0	0	0	0	0	0	0	0	0	0	345	455	24
553043.41		4520	هزینه کل: 557563.41											

جدول (۷): نتایج اجرای روش پیشنهادی با جمعیت‌های مختلف در مثال ۳

زمان	بدترین جواب (\$)	میانگین جواب‌ها (\$)	بهترین جواب (\$)	جمعیت اولیه
75.02	1128049.43	1126043.55	1124423.54	10
220.38	1126868.66	1125507.14	1124377.93	30
372.91	1126218.83	1125310.99	1124293.54	50
482.77	1126270.22	1125207.93	1124293.54	70

جدول (۸): مقایسه نتایج الگوریتم پیشنهادی با سایر روش‌ها در مثال ۳

روش	بهترین جواب (\$)
Lagrange relaxation [۹]	1130660
Integer coded genetic algorithm	1127244
Simulated annealing [۱۲]	1126251
Evolutionary programming [۱۰]	1127257
Genetic algorithm [۹]	1126243
Improved binary particle swarm[۵]	1125216
Extended priority list [۱۸]	1124369
Artificial bee colony	1124293.54

نتیجه‌گیری

برآوردی قیود مسأله که امری مرسوم و البته کندکننده

سرعت همگرایی است، از بین رفته است.

نتایج شبیه‌سازی حاصل از به کارگیری الگوریتم

پیشنهادی گویای همگرایی قابل قبول آن؛ حتی با

تعداد جمعیت‌های اولیه کم است. همچنین، نتایج

مقایسه‌ای با سایر الگوریتم‌های موجود برای حل

مسأله UC نشان می‌دهد که الگوریتم پیشنهادی در

همه مثال‌ها به جواب‌های بهتری در قیاس با سایر

روش‌ها دست یافته است.

در این مقاله، روشی مبتنی بر الگوریتم هوشمند

بهینه‌سازی تجمع زنبور عسل برای حل مسأله UC

پیشنهاد شد. در روش پیشنهادی، محدودیت‌های

حداقل زمان روشن / خاموش به کمک کدبندی عدد

صحیح و محدودیت ذخیره چرخان با استفاده از

کدبندی عدد باینری برآورده می‌شوند. در نتیجه، نیاز

به ضریب جریمه در ارزیابی برازندگی پاسخ‌ها برای

ضمیمه

جدول (۹): مشخصات واحدهای مثال ۱

واحد نیروگاهی										مشخصات
10	9	8	7	6	5	4	3	2	1	واحد
55	55	55	85	80	162	130	130	455	455	P_{max}
10	10	10	25	20	25	20	20	150	150	P_{min}
0.002	0.00222	0.004	0.00079	0.007	0.00398	0.00211	0.002	0.00031	0.00048	a
27.79	27.27	25.92	27.74	22.26	19.7	16.5	16.6	17.26	16.19	b
670	665	660	480	370	450	680	700	970	1000	c
1	1	1	3	3	6	5	5	8	8	MUT
1	1	1	3	3	6	5	5	8	8	MDT
30	30	30	260	170	900	560	550	5000	4500	$H_{start\ up}$
60	60	60	520	340	1800	1120	1100	10000	9000	$C_{start\ up}$
0	0	0	2	2	4	4	4	5	5	T_{cold}
-1	-1	-1	-3	-3	-6	-5	-5	8	8	T^0

جدول (۱۰): میزان تقاضای بار در ساعات مختلف برای مثال ۱

12	11	10	9	8	7	6	5	4	3	2	1	ساعت
1500	1450	1400	1300	1200	1150	1100	1000	950	850	750	700	بار
24	23	22	21	20	19	18	17	16	15	14	13	ساعت
800	900	1100	1300	1400	1200	1100	1000	1050	1200	1300	1400	بار

منابع:

- [10] Kazarlis, S. A., Bakirtzis, A. G. and Petridis, V., "A genetic algorithm solution to the unit commitment problem", *IEEE Trans. on Power Systems*, Vol. 11, No. 1, pp. 83–92, 1996.
- [11] Juste, K. A., Kita, H., Tanaka, E. and Hasegawa, J., "An Evolutionary programming solution to the unit commitment problem", *IEEE Trans. on Power Systems*, Vol. 14, No. 4, pp. 1452–1459, 1999.
- [12] Ebrahimi, J., Hosseinian, S.H. and Gharehpetian, G.B., "Unit commitment problem solution using shuffled frog leaping algorithm", *IEEE Trans. on Power Systems*, Vol. 26, No. 2, pp. 573–581, 2011.
- [13] Simopoulos, D.N., Kavatza, S.D. and Costas D. Vournas, C. D. "Unit Commitment by an Enhanced Simulated Annealing Algorithm ", *IEEE Trans. on Power Systems*, Vol. 21, No.1, pp. 68–76, 2006.
- [14] Sum-im, T. and Ongsakul, W., "Ant colony search algorithm for unit commitment", *Proceedings of IEEE International Conference on Industrial Technology*, pp. 72–77, 2003.
- [15] Jorge, V. and Smith, A., "A seeded memetic algorithm for large unit commitment problems", *Journal of Heuristics*, Vol.8, pp.173–195, 2002.
- [16] Viana, A., Sausa, J. and Matos, M., "Using GRASP to solve the unit commitment problem", *Analysis of Operations Research*, Vol. 120, No. 1, pp.117–132, 2003.
- [17] Balci, H. and Valenzuela, J., "Scheduling electric power generations using particle swarm optimization combined with the Lagrangian relaxation method", *International Journal of Applied Mathematics and Computer Science*, Vol. 14, No. 3, pp. 411–421, 2004.
- [18] Senjyu, T., Miyagi, T., Saber, A.Y., Urasaki, N. and Funabashi, T., "Emerging solution of large-scale unit commitment problem by stochastic priority list", *Electric Power Systems Research*, Vol. 76, No. 5, pp. 283–292, 2006.
- [19] Senjyu, T., Shimabukuro, K., Uezato, K. and Funabashi, T., "A fast technique for unit commitment problem by extended
- [1] نجفی. ا.، فلقی. ح. و فرشاد. م.، " کاربرد الگوریتم تجمع زنبور عسل در مسأله‌ی به مدار آوردن نیروگاه‌ها"، سومین کنفرانس نیروگاه‌های برق، محمودآباد، بهمن‌ماه ۱۳۸۹.
- [2] Yamin, H. Y., "Review on methods of generation scheduling in electric power systems", *Electric Power Systems Research*, Vol. 69, No. 3, pp. 227–248, 2004.
- [3] Ongsaku, W. and Petchara, N., "Unit commitment by enhanced adaptive lagrangian relaxation", *IEEE Trans. on Power Systems*, Vol. 19, No. 1, pp. 620–628, 2004.
- [4] Damousis, I.G., Bakirtzis, A.G. and Dokopoulos, P.S., "A solution to the unit-commitment problem using integer-coded genetic algorithm", *IEEE Trans. on Power Systems*, Vol. 19, No. 2, pp. 1165–1172, 2004.
- [5] Amjady, N. and Shirzadi, A., "Unit commitment using a new integer coded genetic algorithm", *European Trans. on Electrical Power*, Vol. 19, No. 8, pp. 1161–1176, 2009.
- [6] Yuan, X., Nie, H., Su, A., Wanga, L. and Yuan, Y., "An improved binary particle swarm optimization for unit commitment problem", *Expert Systems with Applications*, Vol. 36, No. 4, pp. 8049–8055, 2009.
- [7] Yuan, X., Nie, H., Su, A., Wanga, L. and Yuan, Y., "Application of enhanced discrete differential evolution approach to unit commitment problem", *Energy Conversion and Management*, Vol. 50, No. 9, pp. 2449–2456, 2009.
- [8] Lau, T. W., Chung, C. Y., Wong, K. P., Chung, T. S. and Ho, S. L., "Quantum-inspired evolutionary algorithm approach for unit commitment", *IEEE Trans. on Power Systems*, Vol. 24, No. 3, pp. 1503–1512, 2009.
- [9] Eslamian, M., Hosseinian, S.H. and Vahidi, B., "Bacterial foraging based solution to the unit-commitment problem", *IEEE Trans. on Power Systems*, Vol. 24, No. 3, pp. 1478–1488, 2009.

- [29] Kang, F., Li, J., Li, H., Ma, Z. and Xu, Q., "An improved artificial bee colony algorithm", *Intelligent Systems and Applications*, ISA, 2010.
- [30] Wood, A.J. and Wollenberg, B.F., "Power Generation Operation and Control." New York: Wiley, 1984.

زیر نویس‌ها

1. Unit Commitment (UC)
2. Artificial Bee Colony (ABC)

- priority list", *IEEE Trans. on Power Systems*, Vol. 18, No. 2, pp. 882-888, 2003.
- [20] Gaing, Z. L., "Discrete particle swarm optimization algorithm for unit commitment", *IEEE Power Engineering Society General Meeting*, Vol. 1, Taiwan, 2003.
- [21] Pappala, V.S. and Erlich, I.A., "new approach for solving the unit commitment problem by adaptive particle swarm optimization", *Power and Energy Society General Meeting*, Duisburg, pp. 1-6, 2008.
- [22] Eldin, A.S., El-sayed, M. and Youssef, H., "A two-stage genetic based technique for the unit commitment optimization problem", *12th International Middle East Power System Conference*, MEPCO, Aswan, pp. 425-430, 2008.
- [23] Xiong, W., Li, M.J. and Cheng, Y.L., "An improved particle swarm optimization algorithm for unit commitment", *Proceedings of the 2008 international conference on intelligent computation technology and automation*, Vol. 1, pp. 21-25, 2008.
- [24] Jeong, Y.-W., Park, J.B., Shin, J.R. and Lee, K. Y., "A thermal unit commitment approach using an improved quantum evolutionary algorithm", *Electric Power Components and Systems*, Vol. 37, No. 7, pp. 770-786, 2009.
- [25] Khanmohammadi, S., Amiri, M. and Haque, M. T, " A new three-stage method for solving unit commitment problem", *Energy*, Vol. 35, No. 7, pp. 3072-3080, 2010.
- [26] Patra, S., Goswami, S.K. and Goswami, B., "Fuzzy and simulated annealing based dynamic programming for the unit commitment problem", *Expert Systems with Applications*, Vol. 36, No. 3, pp. 5081-5086, 2009.
- [27] Sishaj, S.H. and Simon, P., "Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions", *Electric Power Components and Systems*, Vol. 38, No. 7, pp. 786- 803, 2010.
- [28] Karaboga, D. and Basturk, B., "On the performance of artificial bee colony (ABC) algorithm", *Applied Soft Computing*, Vol. 8, No. 1, pp. 687-697, 2008.