کشف قوانین جالب از اطلاعات بیولوژیکی با استفاده از الگوریتم ژنتیک موازی

لیلیا پورمدحجی [*,۱]

۱- کارشناس نرم افزار، سازمان آب و برق خوزستان

خلاصه

در این مقاله، یک ژنتیک موازی، بر پایه‌ی روش استخراج قوانین جامع، برای کشف قوانین جالب از یک پایگاه داده بیولوژیکی عظیم پیشنهاد شده است. الگوریتم‌های پیشین و متغیر های آنان برای استخراج قوانین جامع، تکیه بر ۲ پارامتر مرزی، همچون حمایت و توجه حداقل دارند، که موضوعی است که باید مجدداً حل شود. علاوه بر این، موضوعات دیگری، مثل فضای تحقیق وسیع و بهینگی محلی، توجه محققین زیادی را برای استفاده از مکانیزم ابتکاری، به سوی خود جلب می‌کند. در حضور پایگاه‌های داده بیولوژیکی وسیع و با هدف رفع کردن این مشکلات، الگوریتم ژنتیک ممکن است به عنوان یک ابزار مناسب به کار برده شود، اما هزینه‌ی محاسباتی آن، مانع اصلی است. بنابراین، ما الگوریتم‌های ژنتیک موازی را، به منظور آسودگی از مخارج محاسباتی انتخاب می-کنیم. نتیجه‌ی آزمایشگاهی امید بخش است و به تحقیقات بیشتر، مخصوصا در زمینه‌ی علم بیولوژیکی تشویق می‌کند.

کلمات کلیدی: الگوریتم پیشین، الگوریتم ژنتیک، الگوریتم‌های ژنتیک موازی، استخراج قوانین جامع، استخراج اطلاعات

۱. مقدمه

یک ژنتیک موازی مبتنی بر روش استخراج قوانین جامع، به منظور کشف قوانین جالب از یک پایگاه داده بیولوژیکی وسیع یا مجموعه اطلاعات بیومدیکال، توصیه شده است. استخراج قوانین جامع بستگی به ۲ مورد شناخته شده مثل حمایت و توجه دارد. الگوریتم های پیشین برای استخراج قوانین جامع نیز تکیه بر ۲ پارامتر مرزی مثل حمایت و توجه حداقل داشتند. اگرچه، چالش های مشخصی در اعمال الگوریتم های پیشین، مثل الگوریتم حمایت حداقل وابستگی پایگاه داده و فضای تحقیق وسیع وجود دارند. بنابراین، در حضور پایگاه داده های بیولوژیکی وسیع، حدس زدن ارزش مرزی برای حمایت حداقل، کاری مشکل است.

برای جلوگیری از این مشکلات، الگوریتم ژنتیک ممکن است به عنوان یک ابزار مناسب در نظر گرفته شود، اما هزینه‌ی محاسباتی آن مانع اصلی است. بنابراین، ما الگوریتم‌های ژنتیک موازی را، به منظور آسودگی از مخارج محاسباتی انتخاب می‌کنیم. در زمینه‌ی کاری ما، لازم نیست که مقدار حداقل حمایت یا توجه را به کاربر مشخص شده بدهیم، آن به طور خودکار توسط الگوریتم ژنتیک تولید می‌شود. بعضی

[*] **Corresponding author:** کارشناس نرم افزار، سازمان آب و برق خوزستان
Email: pourmodheji.l@kwpa.gov.ir

اوقات، اگر ما آن را توسط کاربر فراهم کنیم، ممکن است که چند الگوی جالب را که ارزش کمتری به عنوان توجه یا حمایت حداقل دارند، از دست بدهیم.

۲. فعالیت‌های اولیه

حداقل حمایت وابستگی پایگاه داده به این معنی است که کاربرها باید مرزهای مناسب را برای فعالیت های استخراجی خود مشخص کنند، اگرچه ممکن است که آنها هیچ دانشی در مورد پایگاه داده‌های خود نداشته باشند. برای جلوگیری از این مشکلات، در این مقاله، ما تمایل داریم که از یک استراتژی استخراجی تکاملی، در هر استخراج قوانین جامع، مبتنی بر یک الگوریتم ژنتیک اجرا شده، استفاده کنیم. مشاهده شده است که ارزیابی صلاحیت در الگوریتم ژنتیک، پر خرج ترین مرحله‌ی آن است؛ بنابراین برای به حداقل رساندن پیچیدگی محاسباتی کلیِ الگوریتم ژنتیک، صلاحیت را به طور موازی محاسبه می‌کنیم. یک مدل در شکل ۲ نشان داده شده است.

الف. استخراج قوانین جامع

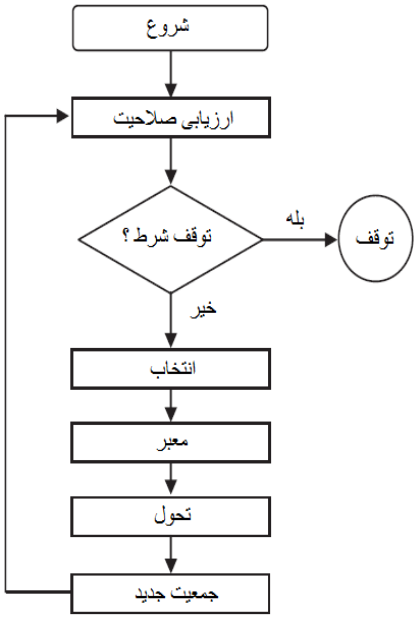
استخراج قوانین جامع، یکی از مهمترین قوانین استخراج اطلاعات است، که برای بسط همبستگی‌های جالب، الگوهای مکرر، و اتحادها در میان مجموعه‌ای از موارد در پایگاه داده‌ی فعل و انفعالی استفاده می‌شوند. به علت درجه‌ی بالای پیاده سازی آن در مناطقی همچون، شبکه های تلکام، مدیریت ریسک، کنترل دارایی و غیره، استخراج قوانین جامع یکی از بهترین تکنیک هایی است که مورد مطالعه قرار گرفته است.

قبل از اینکه ما داخل جزئیات استخراج قوانین جامع شویم، می‌خواهیم که چند اصطلاح پایه ای را تعریف کنیم. اگر ما عالم را به دید مجموعه ای از موارد موجود در مغازه ببینیم، سپس هر مورد یک متغیر بولین دارد، که نشان دهنده‌ی حضور یا عدم حضور آن مورد است. بنابراین هر مجموعه ای از موارد، به عنوان یک مجموعه مورد نامیده می‌شود، که می‌تواند توسط یک بردار بولین از مقادیر تخصیص داده شده به این متغیرها نمایش داده شود. این بردارها می‌توانند برای خریداری الگوهایی که، موارد پرکاربرد خریده می‌شوند، تجزیه و تحلیل شوند. این الگوها می‌توانند در شکل قوانین جامع نمایش داده شوند. بنابراین یک قانون جامع، در مورد رابطه‌ی بین ۲ مجموعه موارد منفصل X و Y است. عبارت $X \rightarrow Y$ بیانگر این است که، هر وقت X اتفاق بیفتد، Y هم اتفاق می‌افتد.

قانون توجه و حمایت، ۲ اندازه‌گیری از میزان جالب بودن یک قانون هستند. حمایت ۲ درصدی بیانگر این است که، خارج از همه‌ی فعل و انفعالات، ۲ درصد نشان می‌دهد که X و Y به همراه یکدیگر خریداری شده‌اند. در حالیکه، توجه ۷۰ درصدی بیانگر این است که، ۷۰ درصد مشتریانی که X را خریداری کرده اند، Y را نیز خریده‌اند. [1]

ب. الگوریتم‌های ژنتیک موازی

الگوریتم ژنتیک (شکل ۱ را ببینید) یک الگوریتم جستجوی ابتکاری است که توسط فرآیند تکاملی توسعه‌ی طبیعی بازرسی می-شود. این ابتکار به طور معمول به منظور کارآمد ساختن راه حل ها برای مشکلات جستجو و بهینه سازی استفاده می‌شود. این به کلاس سطح بالای الگوریتم‌های تکاملی متعلق است (EA).

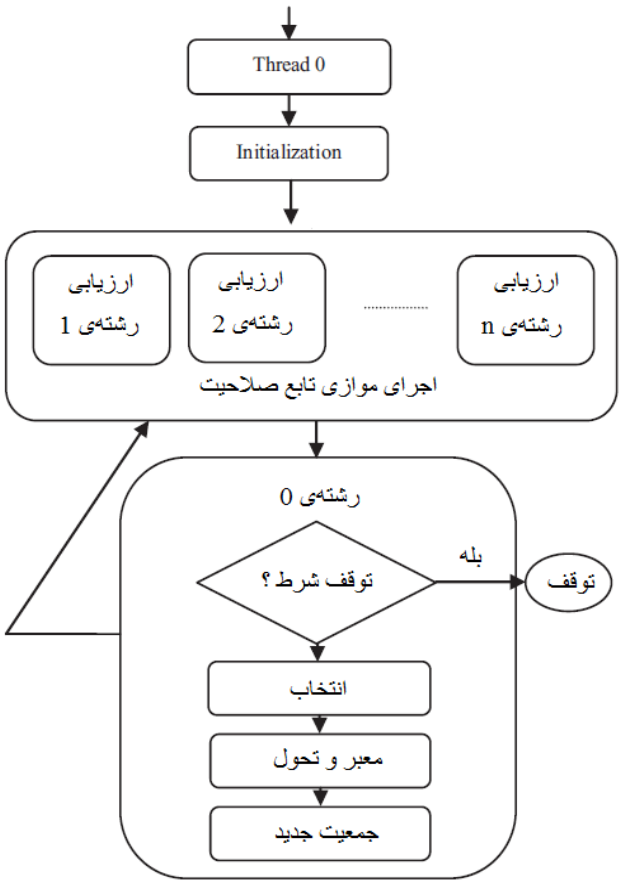


شکل ۱- گراف جریان GA

استاندارد GA در یک روش تکراری، توسط تولید جمعیت های جدید از رشته‌های قدیمی، اجرا می‌شود. هر رشته یک نسخه‌ی رمزگذاری شده از یک راه حل تجربی است. یک تابع ارزیابی، یک اندازه گیری صلاحیت را به هر رشته‌ی مرتبط می‌کند، یک نشان دهنده‌ی تناسب آن به مسئله است. الگوریتم، عملگرهای انتخابی همچون، انتخاب، معبر، و تحول را بر روی یک جمعیت اتفاقی اولیه، به منظور محاسبه‌ی کل نسل رشته های جدید، اعمال می‌کند.[2]

GA های موازی در شکل ۲، تنها نسخه‌ی استاندارد الگوریتم‌های ژنتیک موازی نیستند. در حقیقت، آنها به هدف ایده‌آلِ داشتن الگوریتم موازی‌ای که حضور آن از جمع رفتارهای جدا از زیر الگوریتم‌های اجزای زیر الگوریتم، بهتر است، دست یافته اند، و این دلیل تمرکز ما بر روی آنها است.[3]

یک جمعیت بزرگ در میان تعدادی از گروه‌های نیمه ایزوله توزیع شده اند. یک PGA، مدل گره‌های متصل شده را معرفی می‌کند. انتخاب محلی و قوانین تولید مجدد، به افراد خاص اجازه می‌دهد، که به طور محلی رشد کنند.[4]



شکل ۲- گراف جریان pGA

۳. کار مربوطه

PGA ها الگوریتم‌های احتمالی موازی هستند. همچون الگوریتم‌های ژنتیک سری GA) [5] [6] ها)، آنها مبتنی بر استاندارد تکاملی طبیعی هستند. افراد بهتر جان سالم به در می برند و خود را بهتر از افراد ضعیف بازیابی می‌کنند. برای سرعت بخشیدن به فرآیند تولید جمعیت، ما می‌توانیم جمعیت را به چند زیر جمعیت تقسیم کنیم و آنها را در یک مسیر موازی اجرا کنیم.

سؤالات نظری بسیار مهم در مورد مقایسه‌ی سؤالات کیفیت، بین یک PGA و یک GA کلاسیک مطرح شدند [7]. انبوه جمعیت نیز مهمترین مسأله برای تصمیم‌گیری زمانی است که GA راه حل را پیدا می‌کند.

این بخش چندین پیشرفت جدید را در مطالعه‌ی نظری GA های موازی خلاصه می‌کند. یک مشاهده‌ی مهم از GA های ارباب-برده این است که هر چقدر پردازشگرهای اضافی استفاده می‌شوند، زمان ارزیابی صلاحیت جمعیت کاهش می‌یابد. اما در زمان مشابه، زمان انتقال افراد به برده ها افزایش می‌یابد. توازن بین کم شدن زمان‌های محاسبه و افزایش زمان های ارتباط بیانگر این است که، تعداد بهینه ای از برده ها وجود دارد که زمان کل اجرا را کاهش می‌دهد.

مدل‌های ایده‌آل می‌توانند به چندین روش کامل شوند. استفاده‌ی بیش از تعداد بهینه از گره‌ها خوب نیست، و ممکن است در یک الگوریتم کندتر نمایان شود. GA های موازی سلسله مراتبی می‌توانند به طور موثر، پردازشگرهای بیشتری را مورد استفاده قرار دهند و زمان اجرا را کاهش دهند.

۴. کار پیشنهاد شده

همانطور که می‌دانیم، پارامترهای مشخص شده توسط کاربر، حمایت و توجه حداقل، نقش بسزایی در استخراج قوانین جامع برای یافتن قوانین جالب، ایفا می‌کنند. اما آن تبدیل به یک چالش مهم برای کاربران شده است که پارامتر را در هنگامی که پایگاه داده، شامل تعداد زیادی از رکوردها است، به عنوان توجه و حمایت حداقل شناسایی کنند. هنگامی که پایگاه داده بزرگ است، این نیازمند مطالعه‌ی کامل پایگاه داده برای هر گذر است، که در تعداد زیادی از ورودی/خروجی‌های دیسک، که در نهایت بهره وری الگوریتم را کاهش می‌دهند، نقش دارد.

در این مقاله، ما PGA را به همراه استخراج قوانین جامع به کار بردیم، جاییکه جالب بودن قوانین توسط تابع صلاحیت بدست می‌آید، بنابراین کاربر هرگز نیاز به مشخص کردن پارامتر های اولیه، همچون توجه و حمایت حداقل ندارد.

در تابع صلاحیت، ما توجه مثبت هر مجموعه مورد تکراری را برای یافتن جالب بودن، ارزیابی می‌کنیم. بنابراین، دوره‌ی ارزیابی تابع صلاحیت به تناسب به تعداد رکوردهای حاضر در پایگاه داده مربوط است. بنابراین برای بهینه سازی ارزیابی تابع صلاحیت، ما یک مدل موازی برای ارزیابی تابع صلاحیت به کار بردیم، که به طور ذاتی سری نیست.

الف. رمزگذاری برای GA

ما طرفدار استراتژی میشیگان از رمز گذاری کروموزوم هستیم، جاییکه هر قانون جامع به عنوان یک تک کروموزوم ارائه می‌شود.

C_1	C_2	$C_{...}$	C_m	C_{m+1}	C_{k-1}	C_k

شکل ۳- ارائه‌ی کروموزوم قانون K

ب. عملگرهای GA

برای پیاده سازی الگوریتم ژنتیک، ما ۳ عملگر ژنتیک، مثل، انتخاب، معبر، و تحول را به کار بردیم.

انتخاب تابع (کروموزوم C)، کروموزوم را بر طبق شرایط صلاحیت ارزیابی می‌کند. اگر کروموزوم صلاحیت را داشته باشد مقدار TRUE، و در غیر اینصورت FALSE برمی‌گرداند.

```
Boolean select (Chromosome c)
Begin
        if (fitness(c)>=0) then
                return TRUE;
        else
                return FALSE;
end
```

تغییر دادن تابع (C, pm)، عمل تحول را به طور مناسب در کروموزوم انجام می‌دهد. تابع تصادفی (k+1) یک عدد صحیح تصادفی بین 0 تا k بر می‌گرداند. تابع rand() یک مقدار اعشاری بین ۰ و ۱ بر می‌گرداند.

```
Chromosome mutate (Chromosome c, pm)
begin
        if (rand()<pm) then
        begin
                rand1 ⟵ random(1,k+1)
```

```
            rand2 ⟵ random(1,n+1)
            c[rand1] ⟵ I[rand2];
        end
        return c
end
```

معبر تابع (جمعیت) برای تولید اولاد جدید، با استفاده از جمعیت حاضر، به کار گرفته می‌شود. ما از ۲ استراتژی برای معبر ۲ کروموزوم استفاده کرده‌ایم.

```
Crossover (population, pc)
begin
        populationTemp ⟵ Ø;
        for ∀ Ci = (Ci1, Ci2,…,Cik) population do
        begin
            for ∀ Cj = (Cj1,Cj2,…,Cjk) population and Ci ≠Cj do
            begin
             if (i≠j) then p ⟵ random(k+1)
                    q ⟵ random(k+1)
                    q ⟵ max(p,q)
                    p ⟵ min(p,q)
             C3 ⟵ (Ci1,Ci2,…,Cip,Cip+1,…Cjq,Ciq+1,Cik)
             C4 ⟵ (Cj1,Cj2,…,Cjp,Cip+1,…Ciq,Cjq+1,Cjk)
             populationTemp C3,C4
             end of if
             end of for
             end of While
return populationTemp;
end
```

پ. ارزیابی صلاحیت

تابع صلاحیت نقش بسزایی در تعیین قانون جامع جالب، ایفا می‌کند. آن از شرایط صلاحیت برای خارج کردن کروموزوم جالب استفاده می‌کند.

در الگوریتم فوق الذکر، ما تابع صلاحیت را به عنوان

$$\text{fitness}(c) = \frac{\text{supp}(C1, C2, \ldots, Ck) - \text{supp}(C1, C2, \ldots, Cm)\ \text{supp}(cm+1, \ldots, ck)}{\text{supp}(c1, c2, \ldots, cm)(1 - \text{supp}(cm+1, \ldots, ck))}$$

تعریف می‌کنیم.

ما از محاسبه‌ی موازی برای تابع صلاحیت بالا برای ارزیابی بهینگی زمان محاسبه استفاده می‌کنیم.

ت. جمعیت اولیه

برای پیاده سازی GA، ما نیاز به جمعیت اولیه داریم، که می‌تواند از اعمال مکرر تابع تحول (کروموزوم s، pm) در طول یک دانه تک کروموزوم، مشتق شود.

```
population initialize (Chromosome c)
        begin
                Initialpop[0] ⟵ c;
                while Initialipop[0] < MAXPOPULATION/2 do
                begin
                 populationTemp ⟵ ∅
                   for ∀ c∈ Initialpop[0] do
                 begin
        populationTemp ⟵ populationTemp ∪ mutate (c,1);
                 end
                 Intitalpop[0] ⟵ Intitalpop[0] ∪ populationTemp
                 End
                 Return Initialpop[0]
        end
```

ث. الگوریتم اصلی

جمعیت حاضر به عنوان pop[i] نمایش داده می‌شود. انتخابِ اینجا برای کروموزوم های جالب در جمعیتی که، جمعیت جدید را به عنوان pop[i+1] تولید می‌کند، به کار می‌گیرد. هر جفت از کروموزوم‌ها در جمعیت جدید، به منظور تولید ۲ نسل جدید عبور می‌کنند. این الگوریتم با جمعیتی که شامل کروموزوم های با کیفیت است، مشخص می‌شود.

```
population main (s,ps,p c,pm)
begin
        i ⟵ 0;
        pop[i] ⟵ initialize (c)
while not terminate (pop[i]) do
        begin
                pop[i+1] ⟵ ∅;
                popTemp ⟵ ∅;
                for ∀ c ∈ pop[i] do
                  if select (c,ps) then
                    pop[i+1] ⟵ pop[i+1] ∪ c;
                end
                popTemp ⟵ crossover (pop[i+1],pc);
            for ∀ c ∈ popTemp do
                pop[i+1] ⟵ (pop[i+1] – c) ∪ mutate (c,pm);
                  i ⟵ i+1;
        end
        return pop[i];
end
```

شرایط اتمام الگوریتم مزبور

۱. تفاوت بین بهترین و بدترین کروموزوم از مقدار معین کمتر است.

۲. تعداد تکرار ها از تعداد حداکثر معین "حداکثر حلقه" تجاوز می‌کند.

۵. مطالعه‌ی تجربی و نتایج

ما از الگوریتم مشخص شده در اطلاعات UCI استفاده کردیم. مجموعه اطلاعات شامل ۹۰ رکورد و ۷ ویژگی است.

الف. اطلاعات مجموعه داده

عمل دسته بندی این پایگاه داده، به منظور تعیین منطقه‌ی بازیابی است که باید به مرحله ی بعد ارسال شود. زیرا پایین ترین درجه یک موضوع مهم بعد از عمل است.

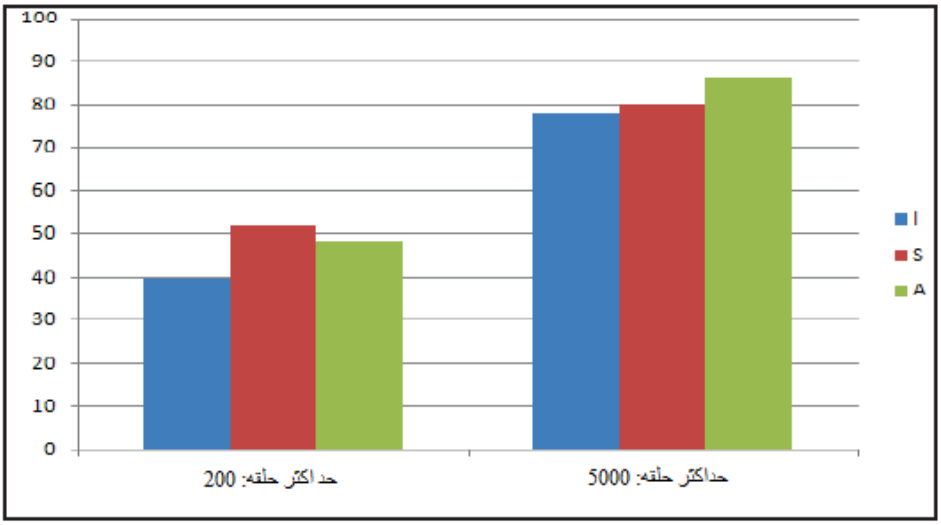
ب. اطلاعات صفت

- L-CORE (patient's internal temperature in C): high (>37), mid (>= 36 and <= 37), low (< 36)
- L-SURF (patient's surface temperature in C): high (>36.5), mid (>= 36.5 and <= 35), low (< 35)
- L-O2 (oxygen saturation in %): excellent (>= 98), good (>= 90 and < 98), fair (>= 80 and <90), poor (<80)
- L-BP (last measurement of blood pressure): high (>130/90), mid (<= 130/90 and >=90/70), low (<90/70)
- SURF-STBL (stability of patient's surface temperature): stable, mod-stable, unstable
- CORE-STBL (stability of patient's core temperature): stable, mod-stable, unstable
- BP-STBL (stability of patient’s blood pressure): stable, mod-stable, unstable

I (patient sent to Intensive Care Unit),
S (patient prepared to go home),
A (patient sent to general hospital floor)

الگوریتم مشخص شده به طور جامع با پارامتر زیر، که در شکل ۳ نشان داده شده است، آزمایش شده است.

حداکثر حلقه ۵۰ ≈ ۵۰۰۰ و مجموعه مقادیر ۰٫۰۳، ۰٫۹۲ ps، ۰٫۸۰ pc، ۰٫۰۲ pm، ۳ k، اندازه‌ی جمعیت مساوی با ۱۰۰، و ما به دقت بالای ۸۵ درصد دست یافتیم.



شکل ۴ – گراف دقت

۶. نتیجه‌گیری

آن تبدیل به یک چالش بزرگ برای کاربر شده است که پارامتر مذکور را، هنگامی که پایگاه داده بسیار بزرگ است، مشخص کند. در این مقاله، ما PGA مبتنی بر استخراج قوانین جامع را پیشنهاد می‌کنیم، جاییکه جالب بودن قوانین، توسط تابع صلاحیت بدست می‌آید و پارامتر مشخص شده توسط کاربر می‌تواند حذف شود.

این کار مجدداً می‌تواند به منظور ایجاد مدل هیبرید، با استفاده از مجموعه‌ی نرم و PGA، برای غلبه بر اطلاعات ناقص برای یافتن قوانین جالب، گسترش داده شود..

۷. قدردانی

در پایان از حمایت مادی سازمان آب و برق خوزستان و دفتر پژوهشهای کاربردی تشکر می‌نمایم.

۸. مراجع

1. J. Han, M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, ISBN 978-81-312-0535-8 2010.

2. T.Bäck, D. Fogel, Z. Michalewicz (eds.) *Handbook of Evolutionary Computation*. Oxford University Press. 1997.

3. E. Cantú-Paz. "*A Summary of Research on Parallel Genetic Algorithms*". R. 95007, July 1995. revised version, IlliGAL R. 97003. May 1997.

4. A. Chipperfield, P. Fleming. "*Parallel Genetic Algorithms*". Parallel and Distributed Computing Handbook, A. Y. H. Zomaya (ed.), MacGraw-Hill, pp. 1118-1143. 1996.

5. A. Grajdeanu. *Parallel Models for Evolutionary Algorithms*. ECLab, George Mason University, 38, 2003.

6. J. J. Grefenstette. *Parallel adaptive algorithms for function optimization*. Report No. CS-81-19, Vanderbilt University, TN, 1981.

7. T. Starkweather, D. Whitley, K. Mathias. *Optimization using distributed genetic algorithms. Parallel Problem Solving from Nature*, Berlin, Germany, 176-185, 1991.

8. E. Cantu-Paze, D. E. Goldberg, "*Modeling Idealized Bounding Cases of Parallel Genetic Algorithms*", Proceedings of the Second Annual Conference, Morgan Kaufmann (San Francisco, CA), 1997.

9. S. Yan, C. Zhang , S. Zhang - ARMGA: *Idetifying Interesting Association Rules with Genetic Algorithm Applied Artificial Intelligence*, 19:677–689 , 2005

10. B. Shumeet, "*The evolution of genetic algorithms: Towards massive parallelism*," in Proceedings of the Tenth International Conference on Machine Learning, San Mateo, CA,, Morgan , pp. 1–8, 1993

11. E. Cant´u-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, 162, 2000.

12. H. Mühlenbein. "*Evolution in Time and Space - The Parallel Genetic Algorithm*". Foundations of Genetic Algorithms, G. J. E. Rawlins (ed.), Morgan Kaufmann, pp. 316-337. 1991.

13. D. E. Goldberg. "*Sizing Populations for Serial and Parallel Genetic Algorithms*". Proceedings of the 3rd ICGA, J. D. Schaffer (ed.), Morgan Kaufmann, pp. 70-79. 1989.