

پردازش تکاملی

پردازش تکاملی Evolutionary Computing

قسمت سوم

دانشگاه صنعتی مالک اشتر

مجتمع دانشگاهی فن آوری اطلاعات و امنیت

زمستان ۱۳۹۲

رقابت بر اساس میزان شایستگی

- انتخاب در دو موقعیت انجام می شود:
 - انتخاب از نسل فعلی برای انتخاب والدینی که قرار است نسل بعد را تولید کند (parent selection)
 - انتخاب از میان والدین + فرزندان برای رفتن به نسل بعد (survivor selection)
- عملگرهای انتخاب بر روی کل افراد اجرا می شود.
 - این عملگرها مستقل از نوع بازنمایی می باشند.
- روشهای انتخاب با یکدیگر در موارد ذیل متفاوتند:
 - عملگرها: احتمالات انتخاب ها را تعریف می کنند.
 - الگوریتم ها: چگونگی پیاده سازی احتمالات را بیان می کنند.

مثال پیاده سازی: SGA

- تعداد مورد انتظار کپی های فرد i

$$E(n_i) = f_i / f_{avg}$$

(μ = pop.size, f_i = fitness of i , f_{avg} avg. fitness in pop.)

- الگوریتم رولت ویل:

- یک توزیع احتمالاتی داریم. برای انجام n انتخاب، یک چرخ یک بازویی را n مرتبه می چرخانیم
- در این روش تضمینی برای مقدار واقعی n_i وجود ندارد.

- الگوریتم بیکرز ساس:

- چرخ می داریم که دارای n بازو به فواصل مساوی است. برای انجام n انتخاب، آن را یکبار می چرخانیم
- این روش تضمین می کند تعداد انتخاب هر فرد i بین کران بالا و پایین تعداد کپی مورد انتظار آن فرد باشد:

$$\text{floor}(E(n_i)) \leq n_i \leq \text{ceil}(E(n_i))$$

انتخاب متناسب با شایستگی (FPS)

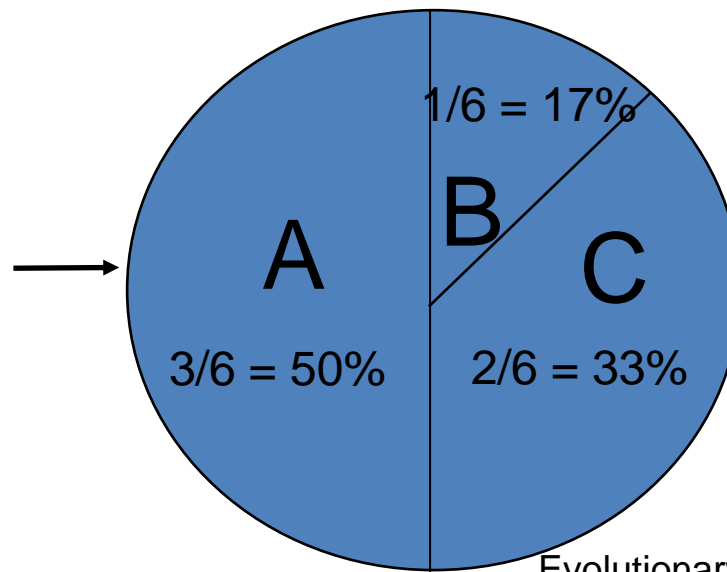
- فرد i دارای احتمال

$$P_i = F_i / \sum F_i$$
 برای انتخاب شدن به عنوان یک والد هست.
- افراد با شایستگی بیشتر احتمال بیشتری برای انتخاب شدن دارند.
- مثال

$$\text{fitness}(A) = 3$$

$$\text{fitness}(B) = 1$$

$$\text{fitness}(C) = 2$$



معایب FPS

- فرض کنید که n نفر باید برای تولید مثل انتخاب شوند.
- تعداد مورد انتظار کپی های هر فرد در استخر جفت گیری برابر است با:
$$E(n_i) = f_i / f_{avg}$$
- بنابراین افراد با شایستگی بالاتر از میانگین تمایل دارند بیشتر از یک کپی در استخر جفت گیری داشته باشند در حالی که افراد دارای تناسب کمتر از میانگین تمایلی به داشتن کپی در استخر ندارند.

معایب FPS

- این مساله منجر به ایجاد مشکلاتی با FPS می شود:
- همگرایی زودرس
- فرض کنید فرد X با $f_{avg} > f_i$ اما $f_i < f_{max}$ در نسل های اولیه تولید شده است. از آنجایی که $n_i \gg 1$ ژن های X به سرعت در سراسر جمعیت گسترش می یابد. در این هنگام، عملگر ترکیب نمی تواند هیچ راه حل جدیدی را تولید کند (بلکه فقط عملگر جهش می تواند) لذا برای همیشه $f_{avg} < f_{max}$ خواهد بود.

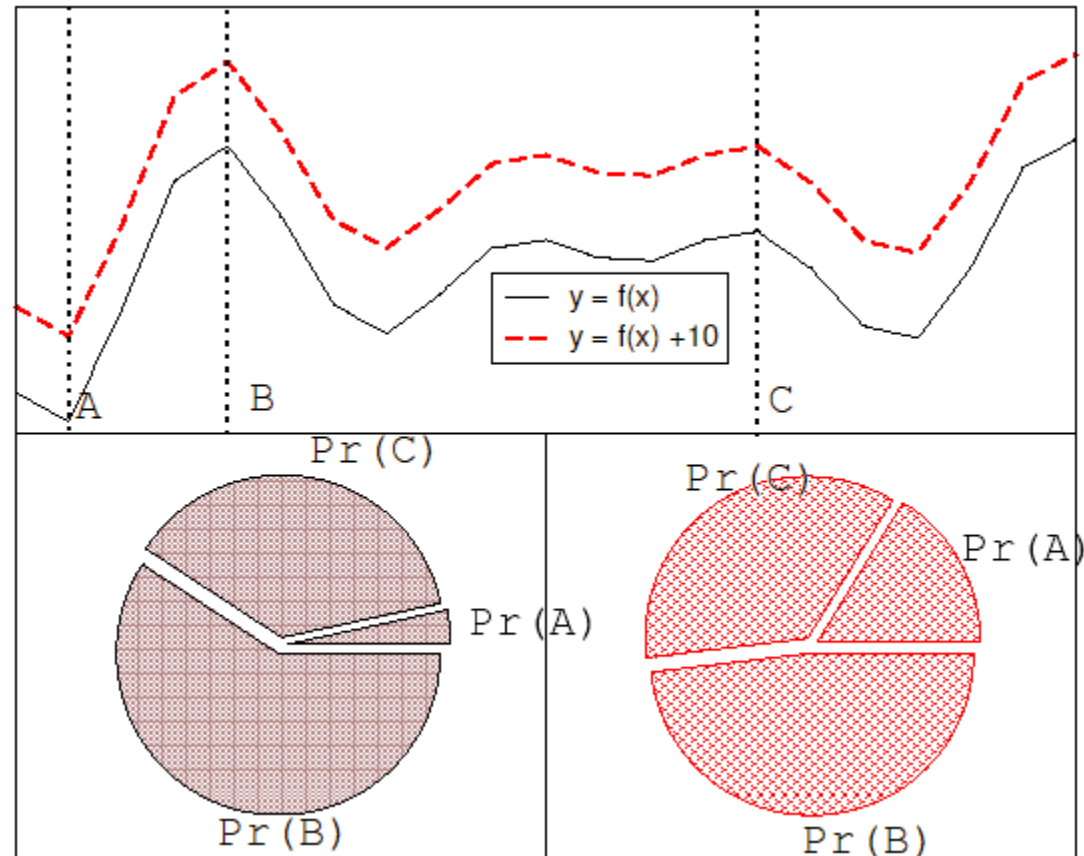
معایب FPS

- ایستایی

- فرض کنید در پایان یک اجرا (یعنی در یکی از نسل های متوالی) همه افراد دارای شایستگی نسبتا بالا و مشابه هستند یعنی برای تمام i ها f_i تقریبا برابر با f_{\max} می باشد.
- بنابراین n_i تقریبا برای تمام i ها ۱ می باشد و عملا هیچ فشار انتخابی وجود ندارد.

- هر دو مشکل (همگرایی زودرس و ایستایی) می تواند با استفاده از تکنیک مقیاس گذاری شایستگی حل شود.

انتقال تابع برای FPS



انتخاب متناسب با شایستگی (FPS)

- مشکلات شامل:

- یک عضو با شایستگی بالا می تواند به سرعت فراگیر شود اگر باقیمانده جمعیت از شایستگی بسیار کمتری برخوردار باشند (همگرایی زودرس)
- در انتهای اجرا زمانی که شایستگی ها مشابه می باشند، فشار انتخاب از بین می رود (ایستایی)

- مقیاس گذاری می تواند دو مشکل اخیر را حل کند.

- **Windowing:** $f'_i = f_i - \beta^t$

- β بدترین شایستگی را در این نسل (یا n نسل اخیر) دارد.

- مقیاس بندی سیگما: $f'_i = \max(f_i - (f_{avg} - c \cdot \sigma_f), 0.0)$
 c یک مقدار ثابت و معمولاً ۲ است.

مقیاس بندی شایستگی

- همانطور که بیان شد دو ویژگی نامطلوب FPS موارد زیر می باشد:
 - همگرایی زود رس: در آغاز تعداد کمی از افراد فوق العاده برای تسلط بر انتخاب می آیند.
 - بعدها فشار انتخابی از بین می رود.
- مقیاس بندی شایستگی، مشکلات فوق را کاهش می دهد

مقیاس بندی

- مقیاس بندی روشی برای کاهش مشکل همگرایی زودرس است.
- در مقیاس بندی خطی $f' = af + b$ می باشد که در آن a و b به گونه ای انتخاب میشوند که بعد از مقیاس بندی، بالاترین شایستگی به اندازه دو برابر میانگین شایستگی ها شود
- در مقیاس بندی سیگما $f' = f - (f_{avg} - c * variance)$ می باشد که در آن برای c عددی بین ۱ تا ۳ انتخاب می شود. شایستگی هایی که حاصل آنها از فرمول فوق مقدار منفی است، صفر در نظر گرفته می شوند.
- در مقیاس بندی توانی $f' = f^k$ می باشد که در آن $k = 1.005$
- مقدار k می تواند با توجه به نوع مساله تغییر نماید

- سه نوع مقیاس بندی عمومی وجود دارد

- مقیاس بندی خطی

- مقیاس بندی سیگما

- مقیاس بندی توانی

- مقیاس بندی خطی

- از رابطه $f' = a.f + b$ برای محاسبه شایستگی جدید استفاده می شود که در آن a و b به گونه ای انتخاب می شوند که

- میانگین شایستگی های جدید برابر با میانگین شایستگی ها قبل از مقیاس بندی است $f'_{avg} = f_{avg}$

- $f'_{max} = c.f_{avg}$ که در آن c برابر است با تعداد کپی های مورد انتظار برای بهترین فرد در جمعیت است. معمولا $c=2$

ملاحظات در مورد مقیاس بندی شایستگی

- یکی از مشکلات مقیاس بندی خطی اینست که تابع مقیاس بندی شده ممکن است مقادیر منفی تولید کند. این اتفاق زمانی رخ می دهد که تعدادی از افراد بد جمعیت دارای شایستگی بسیار پایین تری از f_{avg} باشند و f_{avg} نزدیک به f_{max} باشد.
- یک راه حل برای این مشکل اینست که شایستگی هایی که مقدار آنها منفی میشود را صفر در نظر بگیریم.
- روش دیگر استفاده از مقیاس بندی سیگما است

مقیاس بندی سیگما

- در روش مقیاس بندی سیگما

$$f' = f - (f_{avg} - c \cdot \sigma)$$

- که در آن σ انحراف از معیار جمعیت است و c یک ضریب معقول برای σ می باشد که $1 \leq c \leq 3$
- مقادیر منفی با صفر جایگزین می گردد

مقیاس بندی توانی

- در مقیاس بندی توانی

$$f' = f^k$$

- که باید برای k مقدار مناسبی انتخاب شود
- این مقیاس بندی چندان متداول نیست
- مقدار k بستگی به نوع مسأله دارد. حتی ممکن است نیاز باشد که مقدار آن به صورت پویا در طی ایجاد نسلهای متوالی تغییر کند

عملگرهای انتخاب

Fitness Proportionate Reproduction (Roulette Wheel Selection) •

Random Selection •

Boltzmann Selection •

$$Sel(i, t) = \frac{e^{f_i/T}}{\sum e^{f_i/T}}$$

Rank Selection •

$$Sel(i, t) = \frac{rank(i, t)}{N * (N + 1) / 2}, \text{ where } N \text{ is the best rank}$$

Tournament Selection •

روشهای انتخاب بر مبنای رتبه بندی

- در روشهای انتخاب بر مبنای رتبه بندی، احتمال انتخاب شدن یک فرد بر مبنای رتبه آن فرد در جمعیت تعریف می گردد و نه بر مبنای مقدار شایستگی آن. بدین ترتیب مشکلات روش FPS از بین می رود.
- روش اجرا: ابتدا افراد جمعیت را به ترتیب میزان شایستگی آنها مرتب کرده و سپس به هر یک از آنها امتیازی (احتمال انتخاب شدن) برابر با رتبه شان اختصاص می دهیم. بدین ترتیب بهترین فرد امتیاز μ و بدترین فرد امتیاز ۱ را دریافت می کند.
- این روش دارای سربار مرتب سازی است اما در مقایسه با زمان مورد نیاز برای محاسبه شایستگی افراد جمعیت قابل چشم پوشی است.

انتخاب بر مبنای رتبه بندی

– تمامی افراد جمعیت بر مبنای شایستگی آنها مرتب می شوند.

$$f_j \geq f_i \text{ for all } i < j$$

– به هر فرد احتمال انتخاب شدن p_i را بر مبنای یک توزیع احتمالاتی اختصاص می دهیم

– برخی از توزیع های احتمالاتی:

– توزیع خطی:

$$p_i = a i + b, \text{ with } a < 0 \text{ Negative}$$

– توزیع نمایی:

$$p_i = a \exp(b i + c)$$

رتبه بندی خطی

$$P_{\text{selLR}}(i) = (2 - s)/\mu + 2(i - 1)(s - 1)/(\mu(\mu - 1))$$

- Parameterised by factor s : $1.0 < s \leq 2.0$
 - measures advantage of best individual
 - in SGA this is the number of children allotted to it
- Simple 3 member example

	Fitness	Rank	$P_{\text{sel FPS}}$	$P_{\text{sel LR}} (s=2)$	$P_{\text{sel LR}} (s=1.5)$
A	1	1	0.1	0	0.167
B	5	3	0.5	0.667	0.5
C	4	2	0.4	0.333	0.333
Sum	10		1.0	1.0	1.0

$$P_{exp-rank}(i) = \frac{1 - e^{-i}}{c}$$

Linear Ranking is limited to selection pressure •

Exponential Ranking can allocate more than 2 copies to fittest individual •

Normalise constant factor c according to population size •

Observations on rank selection

Little or no biological justification –
No premature convergence. Because of the ranking –
and probability distribution imposed on it, even less
fit individuals will be selected.

e.g. Let there be three individuals such that $f_1 = 90$,
 $f_2 = 7$, $f_3 = 3$ and $p_i = -0.4i + 1.3$.

With FPS, $p_1 = 0.9 \gg p_2 = 0.07$ and $p_3 = 0.03$, so that
individual 1 comes to saturate the population.

With Rank Selection (RS): —

$p_1 = 0.9$, $p_2 = 0.5$ and $p_3 = 0.1$, so that individual 2 is also selected

No stagnation: •

even at the end, $N_1 \# N_2 \# \dots$ (Similar — arguments to the above)

Explicit fitness values not needed: •

Only the ability to compare solutions —

But, rank selection introduces a reordering • overhead, and makes a theoretical analysis of convergence difficult.

Tournament Selection

- All methods above rely on global population statistics

Could be a bottleneck on parallel machines –

Relies on presence of external fitness function which might not exist: e.g. evolving game players

- Informal Procedure:

Pick k members at random then select the best of these –

Repeat to select more individuals –

Observations on Tournament Selection

Tournament selection inherits the •
advantages of rank selection.

In addition, it does not require global •
reordering and it is more naturally-
inspired.

Tournament selection can be viewed as a •
noisy version of rank selection.

Tournament Selection 2

Probability of selecting i will depend on: •

Rank of i —

Size of sample k —

higher k increases selection pressure •

Whether contestants are picked with replacement —

Picking without replacement increases selection pressure •

Whether fittest contestant always wins —

(deterministic) or this happens with probability p

For $k = 2$, time for fittest individual to take over •

population is the same as linear ranking with $s = 2 \cdot p$

Survivor Selection

Most of methods above used for parent selection •

Survivor selection can be divided into two approaches: •

Age-Based Selection –

e.g. SGA •

In SSGA can implement as “delete-random” •
(not recommended) or as first-in-first-out
(a.k.a. delete-oldest)

Fitness-Based Selection –

Using one of the methods above or •

Two Special Cases

انتخاب بهترین اعضاء

Elitism •

Widely used in both population models (GGA, –
SSGA)

Always keep at least one copy of the fittest –
solution so far

GENITOR: a.k.a. “delete-worst” •

From Whitley’s original Steady-State algorithm (he –
also used linear ranking for parent selection)

Rapid takeover : use with large populations or “no –
duplicates” policy

A.K.A:Also known as

Elitist Selection نخبه گزینی

In elitism ensure that at least one copy of the best individual in the population is always passed onto the next generation. •

The main advantage is that convergence is —
(i.e., if the global maximum is guaranteed discovered, the GA converges to that maximum).

By the same token, however, there is a risk of being —
trapped in a local maximum.

Variation on elitist selection

One alternative is to save the best • individual so far in some kind of register and, at the end of each run, to designate it as the solution instead of using the best of the last generation.

Mapping Quality Functions to Fitness Functions

Background •

So far, we have assumed that there exists a known – quality measure $Q \geq 0$ for the solutions of the problem and that finding a solution can be achieved by maximising Q .

Under this assumption, a chromosome's fitness is – taken to be the quality measure of the individual it encodes.

When this assumption is not valid, adjustments – must be made for fitness-proportionate selection to be used. Of course, one may use a different selection mechanism.

We will look at some of the problems that may – arise with quality measures and suggest ways how these can be mapped into fitness functions that allow FPS to be applied.

Negative-valued Quality Measure

In some problems, Q may take on negative values for some of the solutions •

hence cannot be used directly as a fitness –
function with FPS (it would produce negative probabilities!).

One solution is to use an offset and a threshold –

We can do this by defining fitness as follows: •

and 0 otherwise $f(s) = Q(s) + C_{\min}$ if $Q(s) + C_{\min} > 0$, –

In the above, C_{\min} (the offset) is taken to be one of the –
following:

- 1) The minimum value Q may take (when known) –
- 2) The minimum value of Q in the current and/or last k –
generations
- 3) A function of the variance of Q in the current population, –
e.g. $\text{Mean}(Q) - 2\text{Sqrt}(\text{Variance}(Q))$

Cost-based or Error-based Quality Measure

In some problems, the natural measure of quality is actually a cost or an error E , and finding a solution consists of minimising E (rather than maximising Q). •

In this case, there is a straightforward solution, – which consists of taking $-E$ as the raw fitness, and then using an offset and a threshold (as before) to avoid negative values, if FPS is to be applied.

that is, define f by: •

$$\text{if } C_{\max} > E(s) \quad f(s) = C_{\max} - E(s) \quad -$$

and 0 otherwise

In the above, C_{\max} (the offset) is taken to be one of –
the following:

- 1) The maximum value E may take (when known) –
- 2) The maximum value of E in the current and/or –
last k generations
- 3) A function of the variance of Q in the current –
e.g. $\text{Mean}(E) + 2 \sqrt{\text{Variance}(E)}$ population

Stagnation-prone Quality Measure

If Q ranges over $[Q_{\min}, Q_{\max}]$ where $Q_{\min} \gg 0$ and –
then FPS can lead to stagnation $Q_{\max} - Q_{\min} \gg 0$
even at the beginning of a run.

In this case, one solution is to use an offset and –
threshold as follows:

if $Q(s) + C_{\min} > 0$ $f(s) = Q(s) + C_{\min}$ –
and 0 otherwise,

where: $C_{\min} = -Q_{\min}$,

so that f ranges over $[0, Q_{\max} - Q_{\min}]$

Example application of order based GAs: JSSP

Precedence constrained job shop scheduling problem

J is a set of jobs. •

O is a set of operations •

M is a set of machines •

$Able \subseteq O \times M$ defines which machines can perform which operations •

$Pre \subseteq O \times O$ defines which operation should precede which •

$Dur : \subseteq O \times M \rightarrow \mathbb{R}$ defines the duration of $o \in O$ on $m \in M$ •

The goal is now to find a schedule that is:

Complete: all jobs are scheduled •

Correct: all conditions defined by $Able$ and Pre are satisfied •

Optimal: the total duration of the schedule is minimal •

Precedence constrained job shop scheduling GA

- Representation: individuals are permutations of operations
- Permutations are decoded to schedules by a decoding procedure
 - take the first (next) operation from the individual
 - look up its machine (here we assume there is only one)
 - assign the earliest possible starting time on this machine, subject to machine occupation
- precedence relations holding for this operation in the schedule created so far
- fitness of a permutation is the duration of the corresponding schedule (to be minimized)
- use any suitable mutation and crossover
- use roulette wheel parent selection on inverse fitness
- Generational GA model for survivor selection
- use random initialization

JSSP example: operator comparison

