The (1+λ) Evolutionary Algorithm with Self-Adjusting Mutation Rate

Benjamin Doerr[1] Christian Gießen[2]
Carsten Witt[2] Jing Yang[1]

PGMO DAYS 2017

[1]Laboratoire d'Informatique (LIX)
École Polytechnique
Palaiseau, France

[2]DTU Compute
Technical University of Denmark
Kgs. Lyngby, Denmark

Backgrounds

- $\textsc{OneMax}$ Problem: find a string $x = x_1x_2\dots x_n$ with $x_i \in \{0,1\}$ that minimize

$$\textsc{Om}(x) = \sum_{i=1}^{n} x_i.$$

Backgrounds

- $\textsc{OneMax}$ Problem: find a string $x = x_1 x_2 \ldots x_n$ with $x_i \in \{0, 1\}$ that minimize

$$\textsc{Om}(x) = \sum_{i=1}^{n} x_i.$$

- Black-box complexity (Lehre,Witt (2010)): the smallest expected number of function evaluations needed to solve a problem.

Backgrounds

- $\textsc{OneMax}$ Problem: find a string $x = x_1x_2 \dots x_n$ with $x_i \in \{0, 1\}$ that minimize

$$\textsc{Om}(x) = \sum_{i=1}^{n} x_i.$$

- Black-box complexity (Lehre,Witt (2010)): the smallest expected number of function evaluations needed to solve a problem.
- We only consider unary unbiased variation operators which are symmetric with respect to the bit positions $[n] := \{1, \dots, n\}$ and the bit values 0 and 1.

Structure of (1+λ) EA

Algorithm 1 (1+λ) EA

Select x uniformly at random from $\{0,1\}^n$;
repeat
    for $i = 1$ to λ **do**
        Create x_i by flipping each bit in a copy of x independently with probability r/n;
    end for
    $x^* \leftarrow \arg\min_{x_i} \textsc{Om}(x_i)$;
    if $\textsc{Om}(x^*) \leq \textsc{Om}(x)$ **then**
        $x \leftarrow x^*$
    end if
until $\textsc{Om}(x) = 0$

Previous work and motivation

- Perfomance of EAs on OneMax depends on parameters

Previous work and motivation

- Perfomance of EAs on OneMax depends on parameters
- Classic runtime analysis focus on static parameters (constant mutation probability).

Previous work and motivation

- Perfomance of EAs on $\textsc{OneMax}$ depends on parameters
- Static mutation rate r (i.e. mutation prob. $p = r/n$) for the $(1+\lambda)$ EA (Gießen, Witt, 2016):

$$(1 \pm o(1)) \left(\frac{1}{2} \cdot \frac{n \log \log \lambda}{\log \lambda} + \frac{e^r}{r} \cdot \frac{n \log n}{\lambda} \right)$$

Previous work and motivation

- Perfomance of EAs on $\textsc{OneMax}$ depends on parameters
- Static mutation rate r (i.e. mutation prob. $p = r/n$) for the (1+λ) EA (Gießen, Witt, 2016):

$$(1 \pm o(1)) \left(\frac{1}{2} \cdot \frac{n \log \log \lambda}{\log \lambda} + \frac{e^r}{r} \cdot \frac{n \log n}{\lambda} \right)$$

- Improvement of a factor of $\Theta(\log \log \lambda)$ can be obtained by using dynamic parameter settings

Previous work and motivation

- Perfomance of EAs on $\textsc{OneMax}$ depends on parameters
- Static mutation rate r (i.e. mutation prob. $p = r/n$) for the (1+λ) EA (Gießen, Witt, 2016):

$$(1 \pm o(1)) \left(\frac{1}{2} \cdot \frac{n \log \log \lambda}{\log \lambda} + \frac{e^r}{r} \cdot \frac{n \log n}{\lambda} \right)$$

- Dynamic mutation rate $r = \max \{\ln \lambda / \ln(en/k), 1\}$ where $k = \textsc{Om}(x)$ for the (1+λ) EA (Badkobeh, Lehre, Sudholt, 2014):

$$O\left(\frac{n}{\log \lambda} + \frac{n \log n}{\lambda} \right)$$

best possible among all λ-parallel mutation-based algorithms.

Previous work and motivation

- Perfomance of EAs on $\textsc{OneMax}$ depends on parameters
- Static mutation rate r (i.e. mutation prob. $p = r/n$) for the (1+λ) EA (Gießen, Witt, 2016):

$$(1 \pm o(1)) \left(\frac{1}{2} \cdot \frac{n \log \log \lambda}{\log \lambda} + \frac{e^r}{r} \cdot \frac{n \log n}{\lambda} \right)$$

- Dynamic mutation rate $r = \max\{\ln \lambda / \ln(en/k), 1\}$ where $k = \textsc{Om}(x)$ for the (1+λ) EA (Badkobeh, Lehre, Sudholt, 2014):

$$O\left(\frac{n}{\log \lambda} + \frac{n \log n}{\lambda} \right)$$

best possible among all λ-parallel mutation-based algorithms.

- Mutation rate should depend on the state of the current search process

Previous work and motivation

- Perfomance of EAs on $\textsc{OneMax}$ depends on parameters
- Static mutation rate r (i.e. mutation prob. $p = r/n$) for the (1+λ) EA (Gießen, Witt, 2016):

$$(1 \pm o(1)) \left(\frac{1}{2} \cdot \frac{n \log \log \lambda}{\log \lambda} + \frac{e^r}{r} \cdot \frac{n \log n}{\lambda} \right)$$

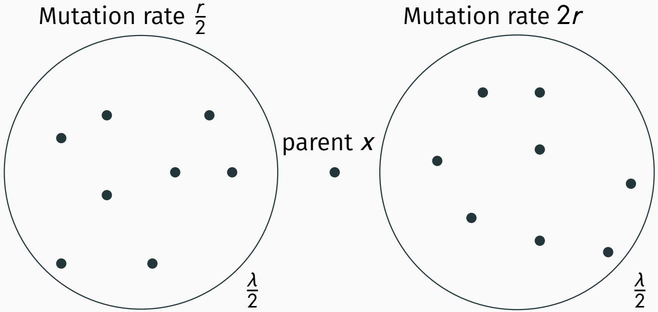
- Dynamic mutation rate $r = \max\{\ln \lambda / \ln(en/k), 1\}$ where $k = \textsc{Om}(x)$ for the (1+λ) EA (Badkobeh, Lehre, Sudholt, 2014):

$$O\left(\frac{n}{\log \lambda} + \frac{n \log n}{\lambda} \right)$$

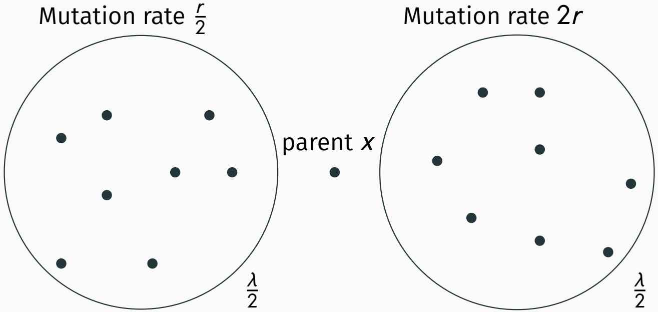
best possible among all λ-parallel mutation-based algorithms.

- Mutation rate should depend on the state of the current search process
- Idea: let the algorithm changes parameters automatically according to recent performance (similar to 1/5-rule)

(1+λ) EA with two-rate standard bit mutation

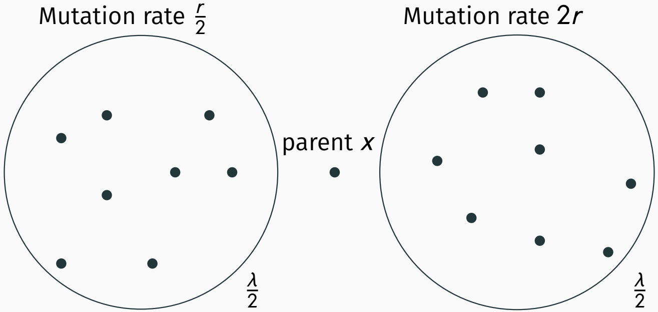


(1+λ) EA with two-rate standard bit mutation



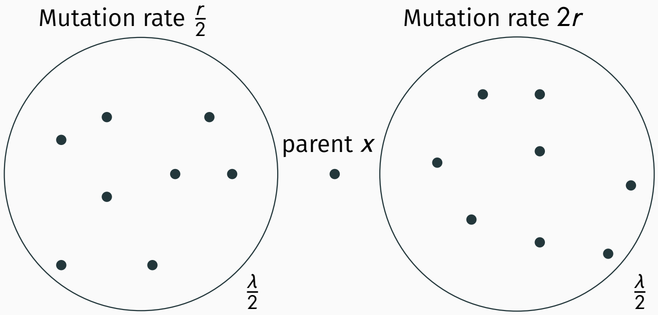
- Replace parent by a best offspring x^* if better or equal (breaking ties randomly, favouring offspring)

(1+λ) EA with two-rate standard bit mutation



- Replace parent by a best offspring x^* if better or equal (breaking ties randomly, favouring offspring)
- Flip a fare coin
  - Heads: Replace r by the rate x^* has been created with
  - Tails: Replace r with $r/2$ or $2r$ with probability $\frac{1}{2}$

(1+λ) EA with two-rate standard bit mutation



- Replace parent by a best offspring x^* if better or equal (breaking ties randomly, favouring offspring)
- Flip a fare coin
  - Heads: Replace r by the rate x^* has been created with
  - Tails: Replace r with $r/2$ or $2r$ with probability $\frac{1}{2}$
- Cap r at 2 and $n/4$

Rough estimation of good r

- Zero-keeping offspring: flip no zero bit, and may flip $0, 1, 2 \cdots$ one bits.

Rough estimation of good r

- Zero-keeping offspring: flip no zero bit, and may flip $0, 1, 2 \cdots$ one bits.
- The probability of keeping all zeros is:

$$\left(1 - \frac{r}{n}\right)^k \geq \left(1 - \frac{r}{n}\right)^n \approx e^{-r}$$

Rough estimation of good r

- Zero-keeping offspring: flip no zero bit, and may flip $0, 1, 2 \cdots$ one bits.
- The probability of keeping all zeros is:

$$\left(1 - \frac{r}{n}\right)^k \geq \left(1 - \frac{r}{n}\right)^n \approx e^{-r}$$

- By taking $r = c \ln(\lambda)$, the zero-keeping population is at least:

$$e^{c \ln \lambda} \cdot \lambda = \lambda^{1-c}$$

Rough estimation of good r

- Zero-keeping offspring: flip no zero bit, and may flip $0, 1, 2 \cdots$ one bits.
- The probability of keeping all zeros is:

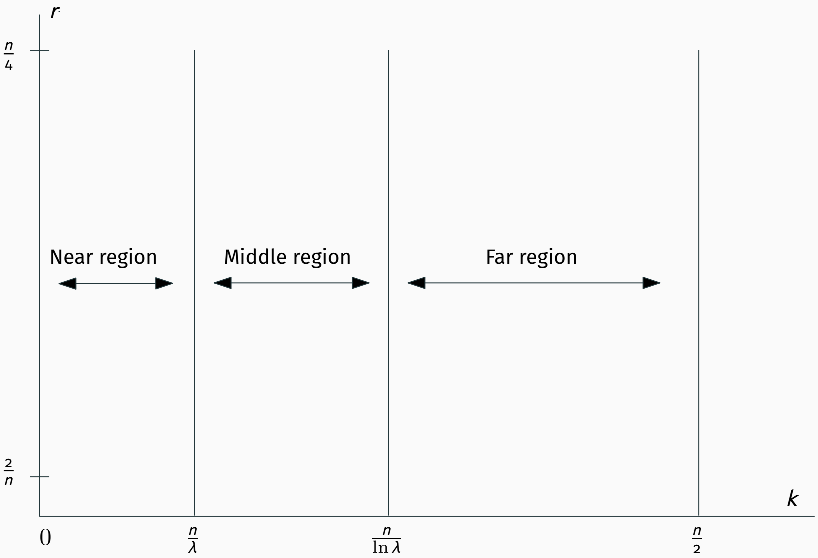
$$\left(1 - \frac{r}{n}\right)^k \geq \left(1 - \frac{r}{n}\right)^n \approx e^{-r}$$

- By taking $r = c\ln(\lambda)$, the zero-keeping population is at least:

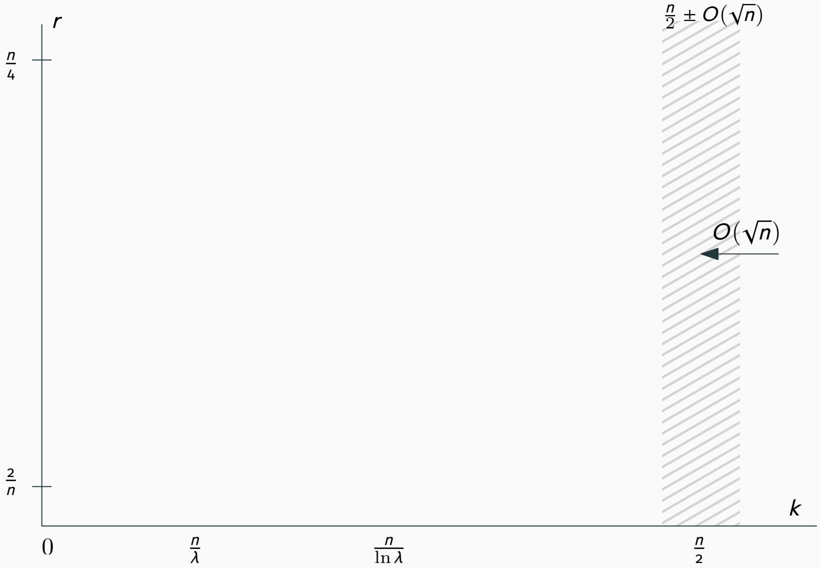
$$e^{c \ln \lambda} \cdot \lambda = \lambda^{1-c}$$

- The best zero-keeping offspring (probably) makes progress.

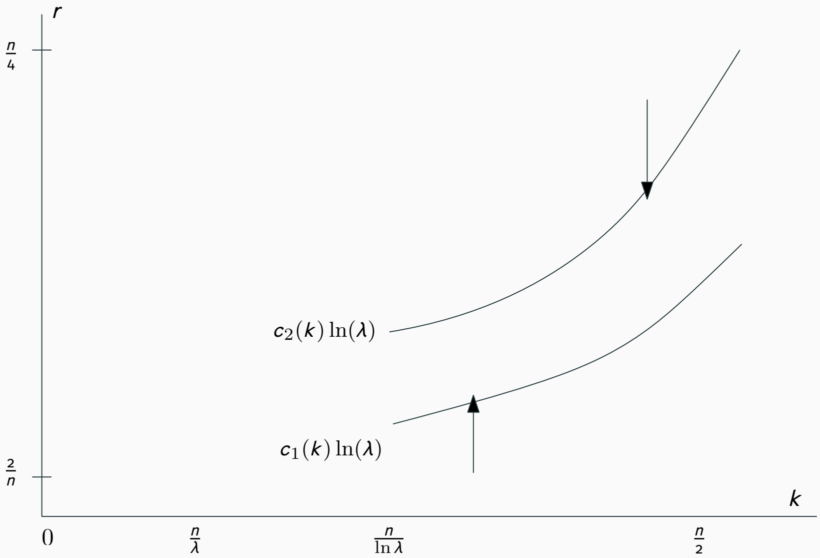
Three regions



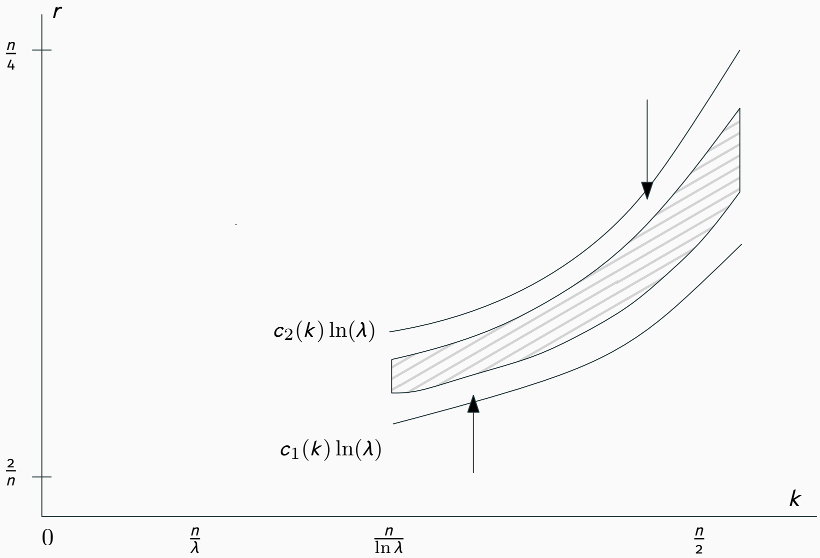
$\sqrt{n}$ drift on distance at the beginning



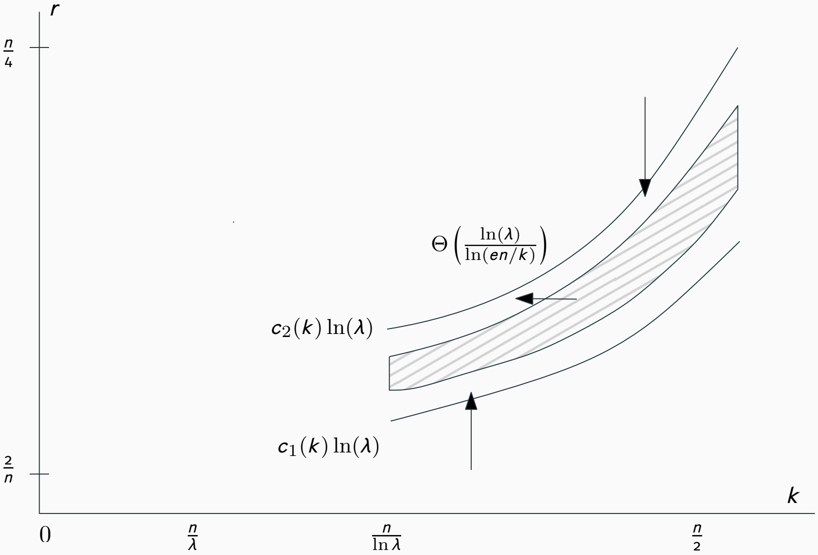
drift on r in far region



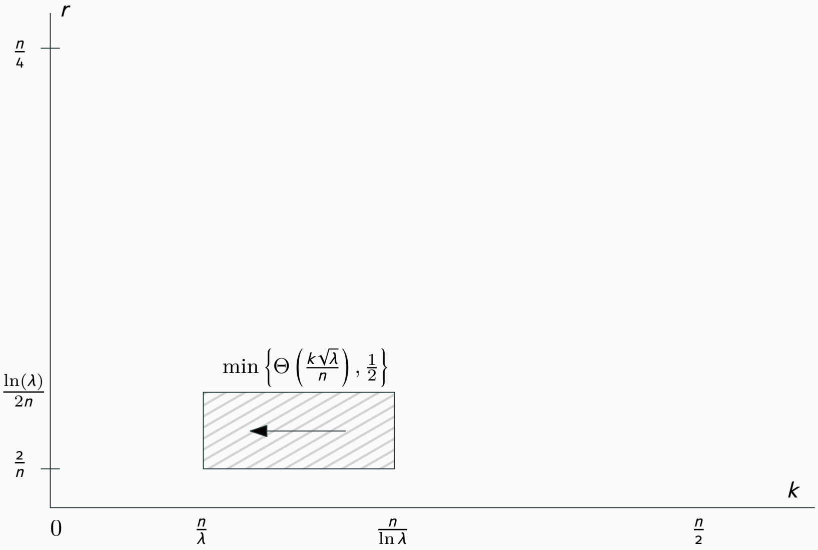
good rate r in far region



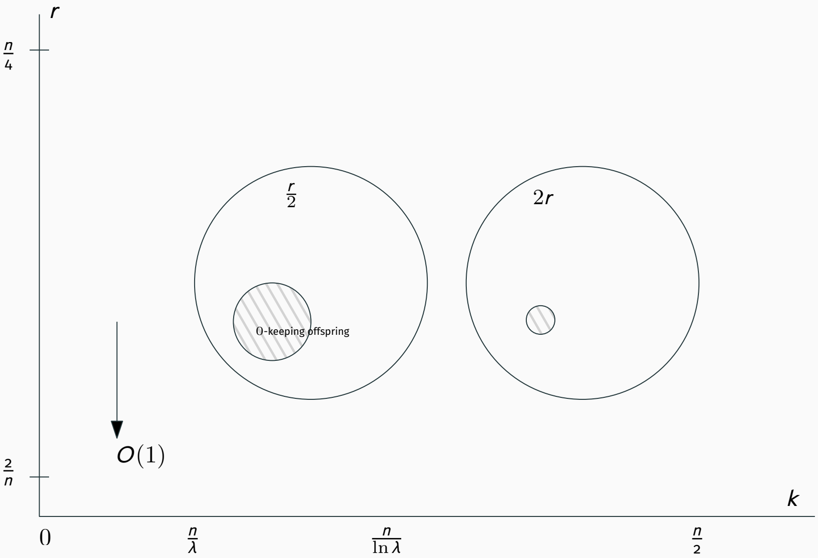
Runtime: $O\left(\frac{n}{\ln(\lambda)}\right)$ generations in far region



Runtime: $O\left(\frac{n}{\ln(\lambda)}\right)$ generations in middle region



Runtime: $O\left(\frac{n\ln(n)}{\lambda}\right)$ generations in far region



Conclusion

- Analysis of a simple self-adjusting mutation scheme for the $(1+\lambda)$ EA

Conclusion

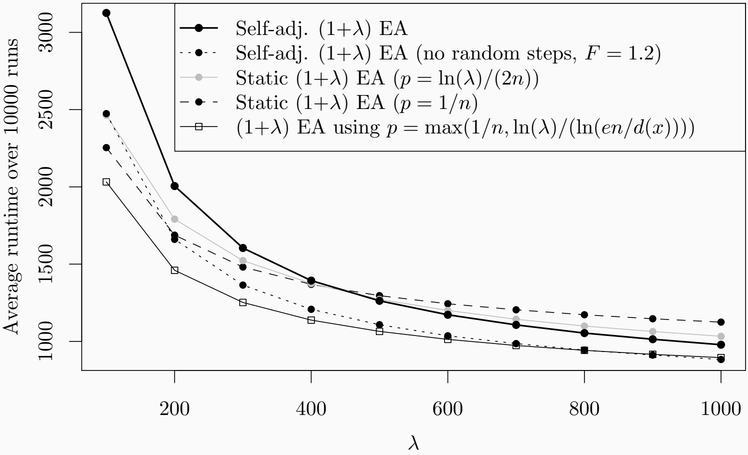
- Analysis of a simple self-adjusting mutation scheme for the $(1+\lambda)$ EA
- Matches the lower bound for every parallel unbiased black-box algorithm

Conclusion

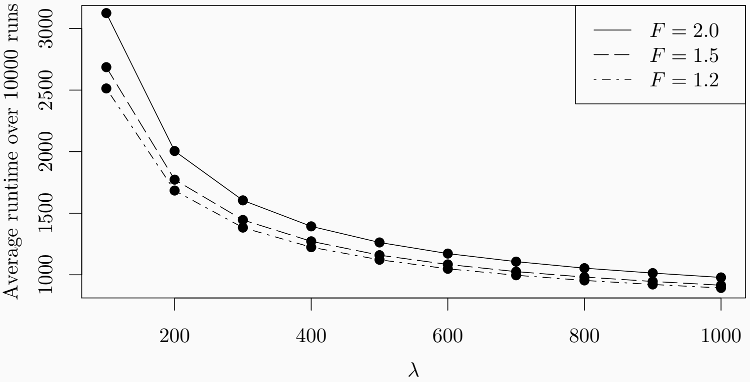
- Analysis of a simple self-adjusting mutation scheme for the $(1+\lambda)$ EA
- Matches the lower bound for every parallel unbiased black-box algorithm
- Side-result: fixed rate of $r = \ln\lambda/2$ yields

$$O(n/\log\lambda + n\log(n)/\sqrt{\lambda})$$

(also optimal for λ not too small)



Self-adj. (1+λ) EA
Self-adj. (1+λ) EA (no random steps, $F = 1.2$)
Static (1+λ) EA ($p = \ln(\lambda)/(2n)$)
Static (1+λ) EA ($p = 1/n$)
(1+λ) EA using $p = \max(1/n, \ln(\lambda)/(\ln(en/d(x))))$
Average runtime over 10000 runs
1000
1500
2000
2500
3000
200
400
600
800
1000
λ



Average runtime over 10000 runs
1000
1500
2000
2500
3000
200
400
600
800
1000
λ
$F = 2.0$
$F = 1.5$
$F = 1.2$