Click to add title

Click to add subtitle

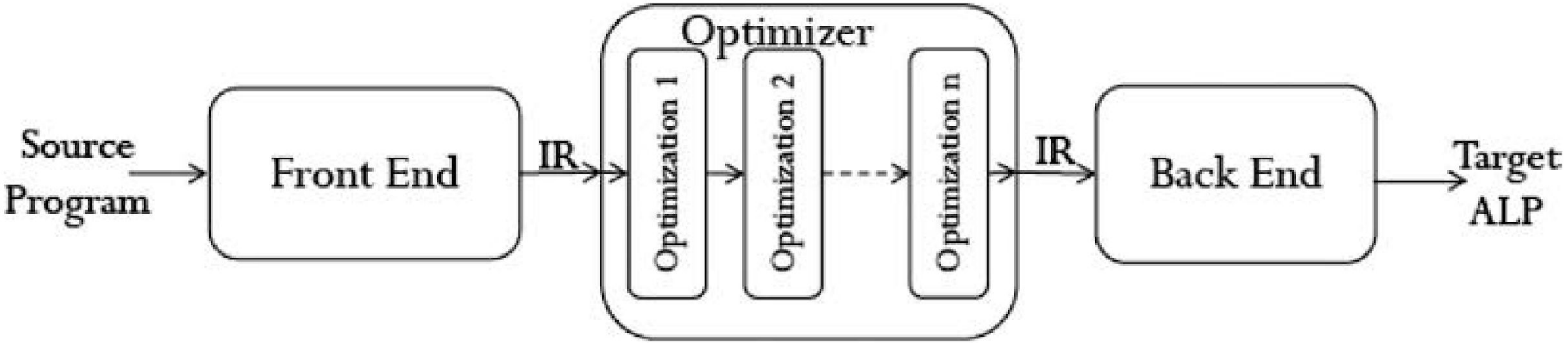
Motivation

- Code size is more important for embedded systems
- Traditional optimizations are designed to minimize execution time
- Developers can tolerate longer compile times
- Interplay between optimizations depends on the program
- Take exponential time to find optimal solution

Why GA is Good?

- Exponential time and space makes it impossible to find optimal solution
- Easy evaluation function
- Objective function is discrete and nonlinear
- Can be run on a program-by-program basis (even module-by-module)

Optimization Passes



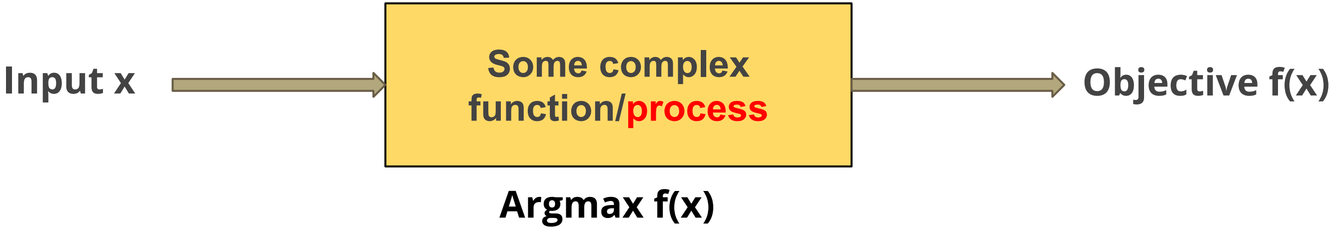
1. Constant propagation (cprop)
2. Dead Code Elimination (dead)
3. Empty Basic Block removal (clean)

Optimization Passes

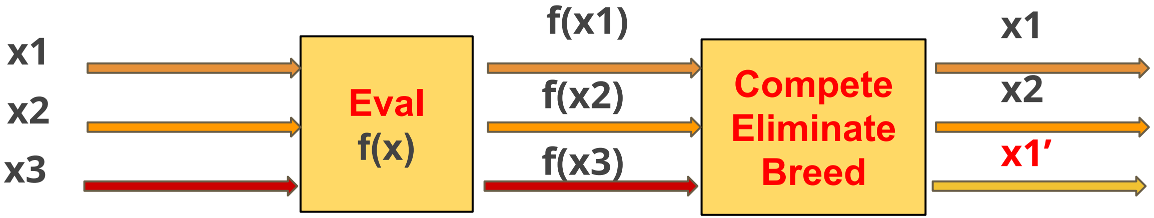
4. Global Value Numbering (valnum)
5. Lazy Code Motion (lazy)
6. Partial Redundancy Elimination (partial)
7. Peephole Optimization (combine)
8. Reassociation (shape)
9. Register Coalescing (coalesce)
10. Operator Strength Reduction (strength)

Genetic algorithm

- Genetic algorithm is a black box optimizer

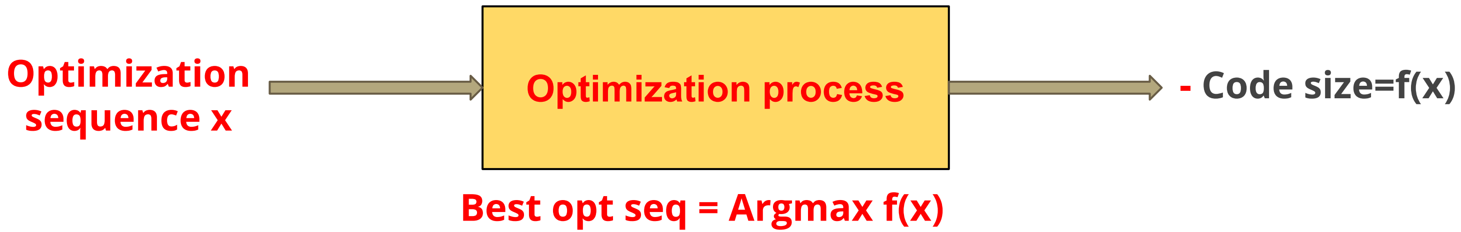


- Genetic algorithm is a search inspired by Darwinian Evolution Principle

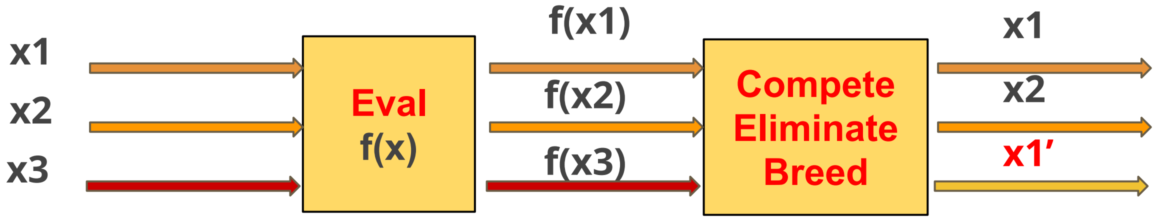


Genetic algorithm

- Genetic algorithm is a black box optimizer



- Genetic algorithm is a search inspired by Darwinian Evolution Principle

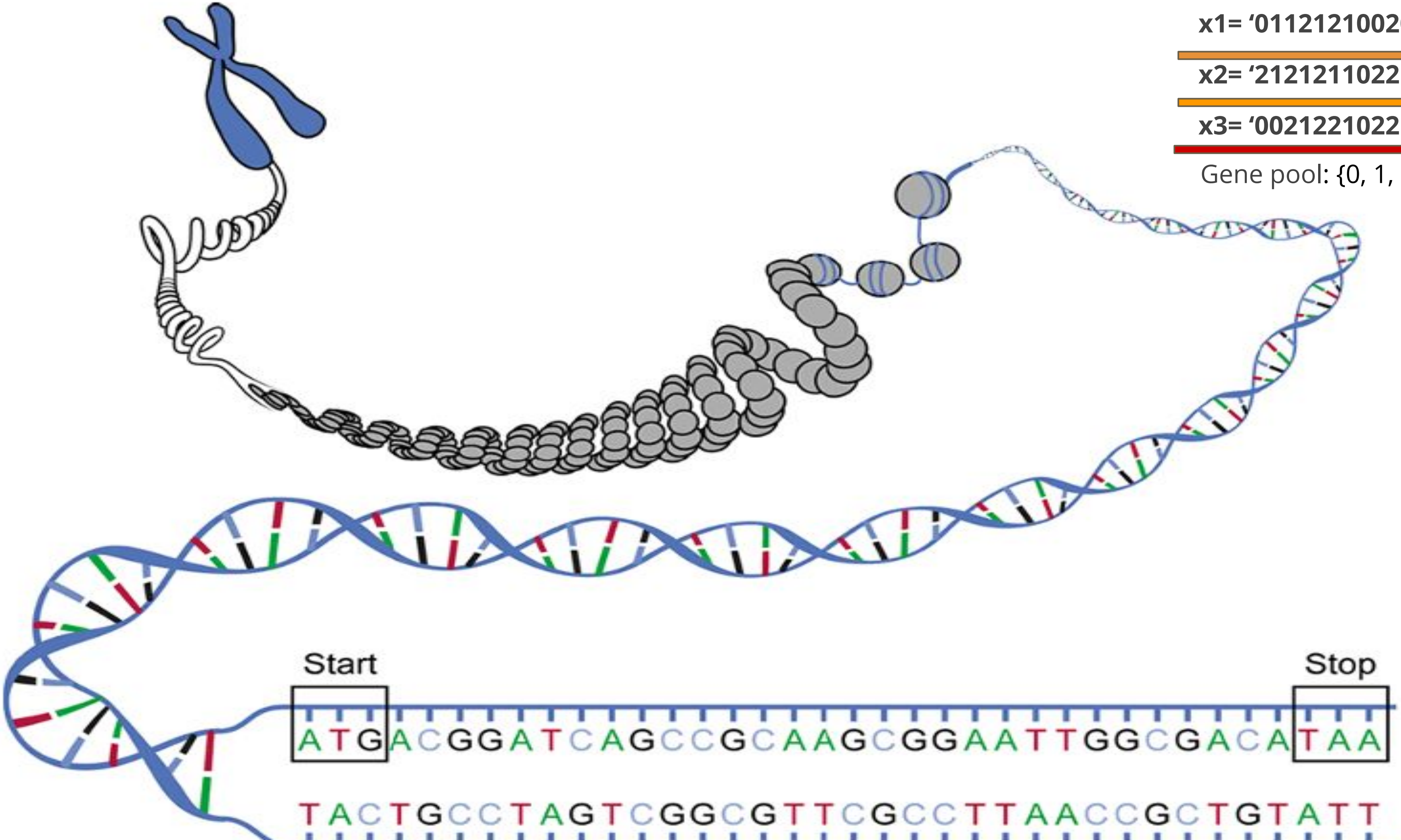


x1= ‘01121210020’

x2= ‘21212110221’

x3= ‘00212210221’

Gene pool: {0, 1, 2}



Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
  - (In this work, gene = opt flag)
  - (In this work, chromosome = opt sequence)
  - L = 12 in this work, i.e. only conisder opt seq with size 12.

x1= ‘01121210020’

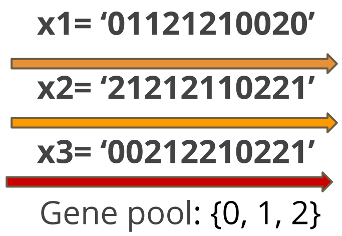
x2= ‘21212110221’

x3= ‘00212210221’

Gene pool: {0, 1, 2}

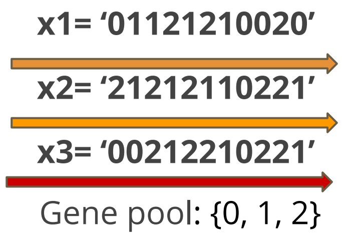
Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.



Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
  - N=20 in this work.



Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
    - A vector with L genes
- Initialize: Randomly generate N chromosomes
    - N chromosomes are called population.
- Genetic algorithm generation: Update the population
    - Evaluation-> Selection->Crossover->Mutation

x1= '01121210020'

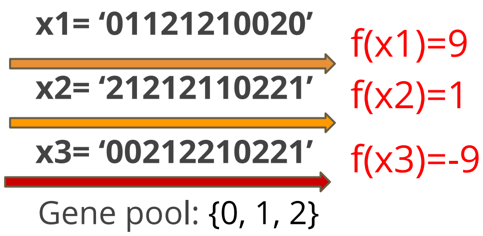
x2= '21212110221'

x3= '00212210221'

Gene pool: {0, 1, 2}

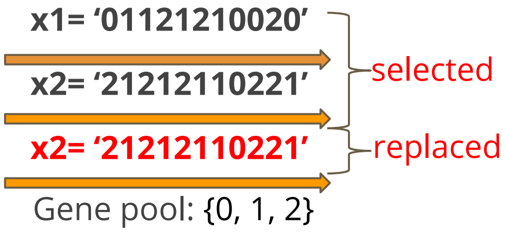
Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - The evaluated objective f(x) is called fitness.



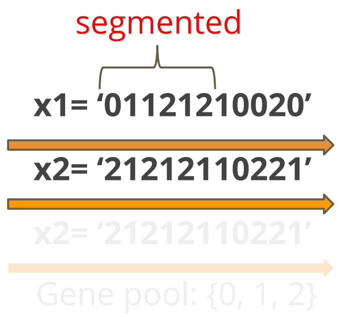
Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - Elicitism: (μ, λ)-selection

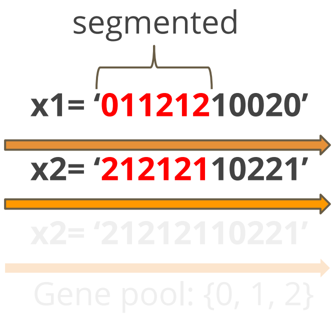


Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - 2-point crossover: Discover new candidate solution from 2 chromosomes

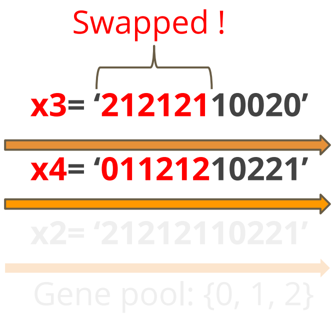


Genetic algorithm: Genetic encoding & Operators



- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - 2-point crossover: Discover new candidate solution from 2 chromosomes

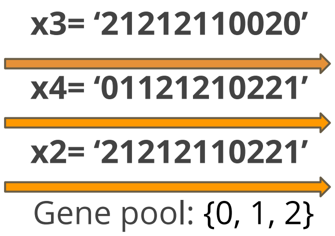
Genetic algorithm: Genetic encoding & Operators



- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - 2-point crossover: Discover new candidate solution from 2 chromosomes

Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - Discover new candidate solution from 1 chromosome.



Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Evaluation-> Selection->Crossover->Mutation
  - Discover new candidate solution from 1 chromosome.
  - Flip genes with some probability.

x3= '20212110120'

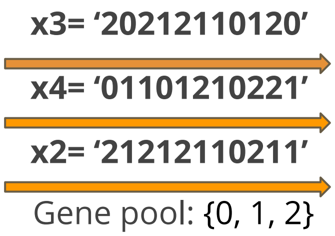
x4= '01101210221'

x2= '21212110211'

Gene pool: {0, 1, 2}

Genetic algorithm: Genetic encoding & Operators

- A solution **x** is represented by a chromosome
  - A vector with L genes
- Initialize: Randomly generate N chromosomes
  - N chromosomes are called population.
- Genetic algorithm generation: Update the population
  - Iterative update until average fitness converges



Genetic algorithm: Why it works

- Problem structure & GA encoding: Let's consider a simple objective function $f(x) = sum(x_i)$, x is binary vector.
    - X1=10011010, f(X1) = 4; X2=11111110, f(X2) = 7

Genetic algorithm: Why it works

- Problem structure & GA encoding: Let's consider a simple objective function $f(x) = sum(x_i)$, x is binary vector.
  - X1=10011010, f(X1) = 4; X2=11111110, f(X2) = 7
  - Segment '..1..' is a good schemata that need to be preserved to form a better candidate -> Selection + Crossover !

Experiment

- Fortran
  - fmin, rkf45, seval, solve, svd, urand, zeroin
- C code
  - adpcm, compress, fft, dfa, dhrystone, nsieve
- 1000 generations / **1 day**

Optimizing with GA

	Unoptimized		GA results				
Benchmark	static	dynamic	static	% red.	dynamic	% red.	gen. found
adpcm	438	17221981	351	19.9	12290460	28.6	6
compress	1753	8402188	1318	24.8	5545480	34.0	(77,79)
dfa	1744	842382	1107	36.5	496164	41.1	806
dhrystone	760	4920264	536	29.5	3200191	35.0	(22,920)
fft	2415	18339859	1757	27.2	14574279	20.5	2
fmin	374	2192	187	50.0	963	56.1	32
nsieve	353	761244374	202	42.8	539954218	29.1	(0, 189)
rkf45	1525	511251	745	51.1	201470	60.6	74
seval	1061	3594	288	72.9	842	76.6	39
solve	1023	2729	437	57.3	1029	62.3	(33,58)
svd	2087	13049	972	53.4	4760	63.5	26
tomcatv	2250	1379982621	551	75.5	232833969	83.1	90
urand	204	1563	93	54.4	613	60.1	(0,18)
zeroin	273	1815	150	45.1	809	55.4	(239,270)

Optimizing with GA

Static operation

	Unoptimized		GA results				
Benchmark	static	dynamic	static	% red.	dynamic	% red.	gen. found
adpcm	438	17221981	351	19.9	12290460	28.6	6
compress	1753	8402188	1318	24.8	5545480	34.0	(77,79)
dfa	1744	842382	1107	36.5	496164	41.1	806
dhrystone	760	4920264	536	29.5	3200191	35.0	(22,920)
fft	2415	18339859	1757	27.2	14574279	20.5	2
fmin	374	2192	187	50.0	963	56.1	32
nsieve	353	761244374	202	42.8	539954218	29.1	(0, 189)
rkf45	1525	511251	745	51.1	201470	60.6	74
seval	1061	3594	288	72.9	842	76.6	39
solve	1023	2729	437	57.3	1029	62.3	(33,58)
svd	2087	13049	972	53.4	4760	63.5	26
tomcatv	2250	1379982621	551	75.5	232833969	83.1	90
urand	204	1563	93	54.4	613	60.1	(0,18)
zeroin	273	1815	150	45.1	809	55.4	(239,270)

Optimizing with GA

Dynamic operation

	Unoptimized		GA results				
Benchmark	static	dynamic	static	% red.	dynamic	% red.	gen. found
adpcm	438	17221981	351	19.9	12290460	28.6	6
compress	1753	8402188	1318	24.8	5545480	34.0	(77,79)
dfa	1744	842382	1107	36.5	496164	41.1	806
dhrystone	760	4920264	536	29.5	3200191	35.0	(22,920)
fft	2415	18339859	1757	27.2	14574279	20.5	2
fmin	374	2192	187	50.0	963	56.1	32
nsieve	353	761244374	202	42.8	539954218	29.1	(0, 189)
rkf45	1525	511251	745	51.1	201470	60.6	74
seval	1061	3594	288	72.9	842	76.6	39
solve	1023	2729	437	57.3	1029	62.3	(33,58)
svd	2087	13049	972	53.4	4760	63.5	26
tomcatv	2250	1379982621	551	75.5	232833969	83.1	90
urand	204	1563	93	54.4	613	60.1	(0,18)
zeroin	273	1815	150	45.1	809	55.4	(239,270)

Optimizing with GA

# of generation

	Unoptimized		GA results				
Benchmark	static	dynamic	static	% red.	dynamic	% red.	gen. found
`adpcm`	438	17221981	351	19.9	12290460	28.6	6
`compress`	1753	8402188	1318	24.8	5545480	34.0	(77,79)
`dfa`	1744	842382	1107	36.5	496164	41.1	806
`dhrystone`	760	4920264	536	29.5	3200191	35.0	(22,920)
`fft`	2415	18339859	1757	27.2	14574279	20.5	2
`fmin`	374	2192	187	50.0	963	56.1	32
`nsieve`	353	761244374	202	42.8	539954218	29.1	(0, 189)
`rkf45`	1525	511251	745	51.1	201470	60.6	74
`seval`	1061	3594	288	72.9	842	76.6	39
`solve`	1023	2729	437	57.3	1029	62.3	(33,58)
`svd`	2087	13049	972	53.4	4760	63.5	26
`tomcatv`	2250	1379982621	551	75.5	232833969	83.1	90
`urand`	204	1563	93	54.4	613	60.1	(0,18)
`zeroin`	273	1815	150	45.1	809	55.4	(239,270)

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	rnldoncvconv	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	rnldoncvconv	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvds$\boxed{\text{nnn}}$ss	odvs
tomcatv	dntvv$\boxed{\text{cc}}$ocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

duplicated?

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	rnldoncvconv	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

rnldoncvconv

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	$\boxed{\texttt{rnldoncvconv}}$	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

rnldoncvconv

12

nldoncvconv
rldoncvconv
rndoncvconv
.
.
.
rnldoncvcnv
rnldoncvcov
rnldoncvcon

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	$\boxed{\texttt{rnldoncvconv}}$	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

rnldoncvconv

12

nldoncvconv
rldoncvconv
~~rndoncvconv~~
.
.
.
~~rnldoncvcnv~~
rnldoncvcov
rnldoncvcon

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	rnldoncvconv	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

rnldoncvconv

nldoncvconv
rldoncvconv
~~rndoncvconv~~
.
.
.
~~rnldoncvcnv~~
rnldoncvcov
rnldoncvcon

...

Optimization Sequence found by GA

Benchmark	GA sequence	reduced sequence
adpcm	tonosdnzscno	osnzc
compress	ncsnzosvndvs	nczvnds
dfa	rztonvncodvs	rztcodvs
dhrystone	covolcnocdvc	covlcodvc
fft	ovocdvscnnos	ovcdvs
fmin	tdcotcscnnvo	dcsvo
nsieve	snzdvvcdsvss	nzdvcs
rkf45	ovndtdcvsndd	ovndtcvsd
seval	rnldoncvconv	ldovco
solve	dondvnsdsdcv	ovndcv
svd	odvdvdsnnnss	odvs
tomcatv	dntvvccocvdv	tvcodv
urand	cnottcdtvooc	nodvc
zeroin	rsdosvncnsss	rsosvcs

gene	optimization
c	cprop
d	dead
l	partial
n	clean
o	combine
r	shape
s	coalesce
t	strength
v	valnum
z	lazy

rnldoncvconv

nldoncvconv
rldoncvconv
~~rndoncvconv~~
.
.
.
~~rnldoncvcnv~~
rnldoncvcov
rnldoncvcon

...

GA against fixed sequence

- System default that produces low ***dynamic*** operation counts
  - *rvzcodtvzcod*
- Observations of optimization sequences returned by the GA
  - *nodvcodvs*

GA against fixed sequence

	rvzcodtvzcod		nodvcodvs	
Benchmark	operations	%	operations	%
adpcm	362	3.0	356	1.4
compress	1412	6.7	1325	0.5
dfa	1168	5.2	1145	3.3
dhrystone	574	6.6	544	1.5
fft	1973	10.9	1757	0.0
fmin	203	7.9	198	5.6
nsieve	227	11.0	202	0.0
rkf45	832	10.5	751	0.8
seval	313	8.0	297	3.0
solve	609	28.2	438	0.2
svd	1641	40.8	973	0.1
tomcatv	770	28.4	565	2.5
urand	93	0.0	93	0.0
zeroin	158	5.1	154	2.6

	rvzcodtvzcod		nodvcodvs	
Benchmark	operations	%	operations	%
adpcm	11965360	-2.7	12440360	1.2
compress	7494982	26.0	5548268	0.1
dfa	546396	9.2	511349	3.0
dhrystone	3390200	5.6	3270193	2.1
fft	15088485	3.4	14574279	0.0
fmin	955	-0.8	947	-1.7
nsieve	570608938	5.4	554450986	2.6
rkf45	214146	5.9	215620	6.6
seval	809	-4.1	985	14.5
solve	1099	6.4	1033	0.4
svd	4731	-0.6	4758	-0.0
tomcatv	186355888	-24.9	239463612	2.8
urand	631	2.9	613	0.0
zeroin	829	2.4	811	0.2

GA against fixed sequence

	rvzcodtvzcod		nodvcodvs	
Benchmark	operations	%	operations	%
adpcm	362	3.0	356	1.4
compress	1412	6.7	1325	0.5
dfa	1168	5.2	1145	3.3
dhrystone	574	6.6	544	1.5
fft	1973	10.9	1757	0.0
fmin	203	7.9	198	5.6
nsieve	227	11.0	202	0.0
rkf45	832	10.5	751	0.8
seval	313	8.0	297	3.0
solve	609	28.2	438	0.2
svd	1641	40.8	973	0.1
tomcatv	770	28.4	565	2.5
urand	93	0.0	93	0.0
zeroin	158	5.1	154	2.6

	rvzcodtvzcod		nodvcodvs	
Benchmark	operations	%	operations	%
adpcm	11965360	-2.7	12440360	1.2
compress	7494982	26.0	5548268	0.1
dfa	546396	9.2	511349	3.0
dhrystone	3390200	5.6	3270193	2.1
fft	15088485	3.4	14574279	0.0
fmin	955	-0.8	947	-1.7
nsieve	570608938	5.4	554450986	2.6
rkf45	214146	5.9	215620	6.6
seval	809	-4.1	985	14.5
solve	1099	6.4	1033	0.4
svd	4731	-0.6	4758	-0.0
tomcatv	186355888	-24.9	239463612	2.8
urand	631	2.9	613	0.0
zeroin	829	2.4	811	0.2

Conclusion

- Program-specific optimization sequence
- Reduction of code size up to 40%

Discussion

- Compare to known optimization flag O2, O3
- Convergence too slow
- Different fitness function
    - Execution time, power consumption