

Usporedba evolucijskih algoritama



Motivacija

- Kako odabrati prikladni algoritam za problem?
- Kako podesiti parametre za algoritme?
- Kako usporediti višekriterijske algoritme?

Usporedba algoritama

- Pokušaj:
 - Izvedi svaki algoritam jednom na nekom problemu
 - Zabilježi najbolje dobiveno rješenje
 - Najbolji je onaj koji je postigao najbolje rješenje
- Ispravno jedino ako su algoritmi deterministični
- Evolucijski algoritmi su uglavnom stohastički
 - Nije prikladno jer se rezultat lako može promijeniti nakon ponovnog izvođenja

Usporedba algoritama

- Kod usporedbe stohastičkih algoritama, potrebno je nekoliko puta ponoviti eksperiment
- Koliko? Što više to bolje
 - Obično 30-50 puta
- Na temelju dobivenih rezultata određuje se koji su rezultati najbolji
 - Kako?

Kriterij zaustavljanja

- Koji kriterij zaustavljanja koristiti?
 - Kriterij koji osigurava da su algoritmi odredili jednak posao
- Broj iteracija nije dobar -> o algoritmu ovisi što se dešava u jednoj iteraciji
- Koristiti:
 - Maksimalni broj evaluacija funkcije cilja
 - Vremensko ograničenje

Statistički pokazatelji

- Minimum
- Srednja vrijednost
- Medijan
- Maksimum
- Standardna devijacija
- ...

Koju mjeru koristiti za usporedbu

- Minimum?
 - Pokazuje najbolje što smo dobili algoritmom
 - Može biti outlier i možda smo samo slučajno dobili taj rezultat
 - Nije dobar za usporedbu

Srednja vrijednost i medijan

- Obično se koriste za usporedbu jer opisuju prosječni rezultat
- Koju mjeru koristiti?
 - Ovisi o rezultatima

Srednja vrijednost i medijan

- Srednja vrijednost:
 - Najčešće korištena
 - Koristi se za izračun standardne devijacije
 - Jako varira ovisno o outlierima
 - Pogodna za podatke s normalnom distribucijom
- Medijan:
 - Predstavlja sredinu podataka (ispod te vrijednosti nalazi se 50% podataka, isto kao i iznad nje)
 - Otpornija na outliere

Srednja vrijednost i medijan

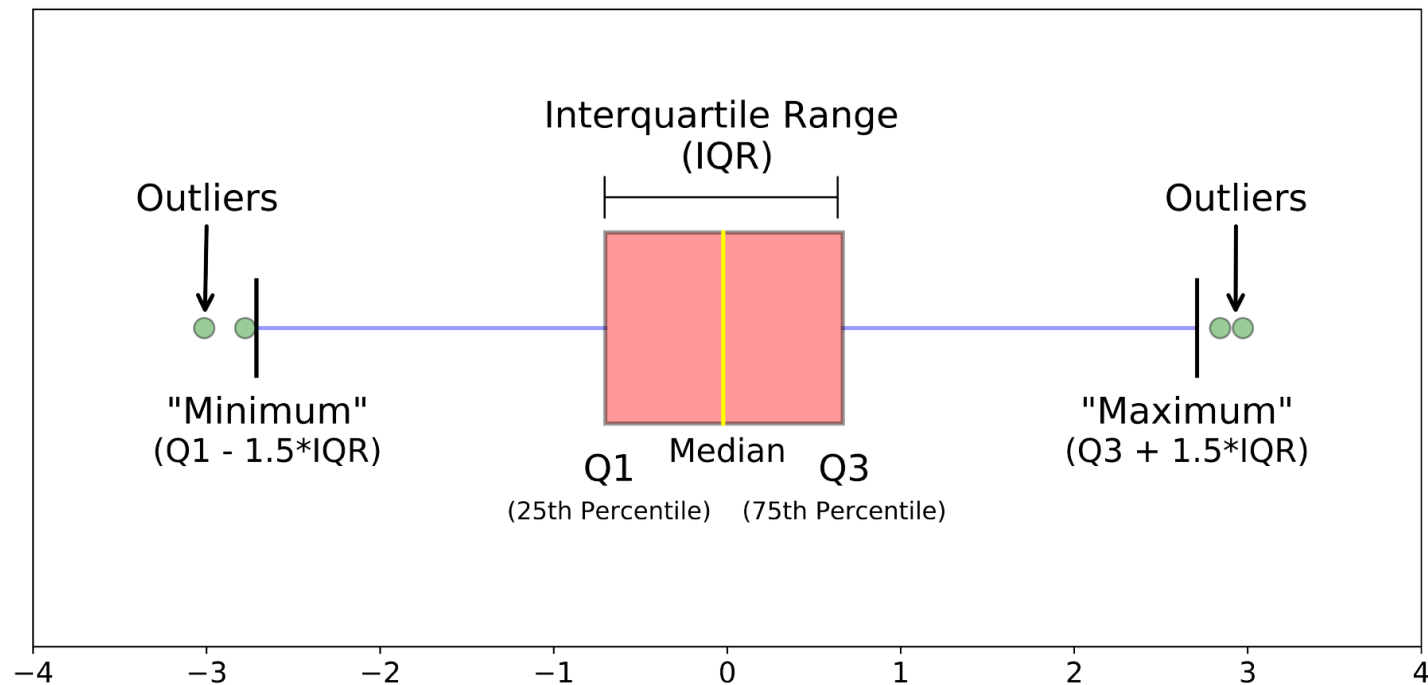
	Eksperiment 1	Eksperiment 2
	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
	7	70
Srednja vrijednost	4	13
Medijan	4	4

Standardna devijacija

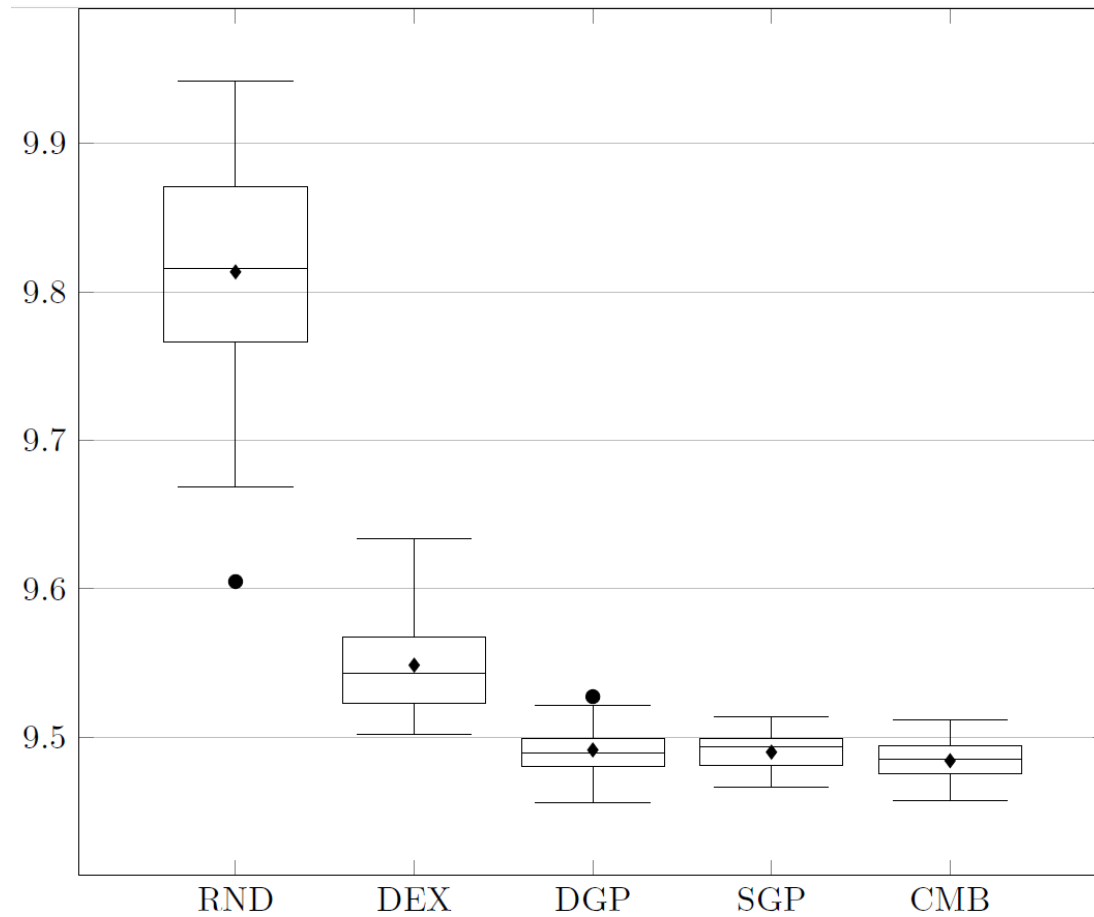
- Označava raspršenost rješenja
 - Velika vrijednost -> rješenja raspršena
 - Mala vrijednost -> rješenja grupirana oko srednje vrijednosti
- Obično se koristi uz srednju vrijednost
 - Odnosno za normalno distribuirane podatke

Grafički prikaz rezultata

- Box plot – prikazuje okvirnu distribuciju dobivenih rješenja

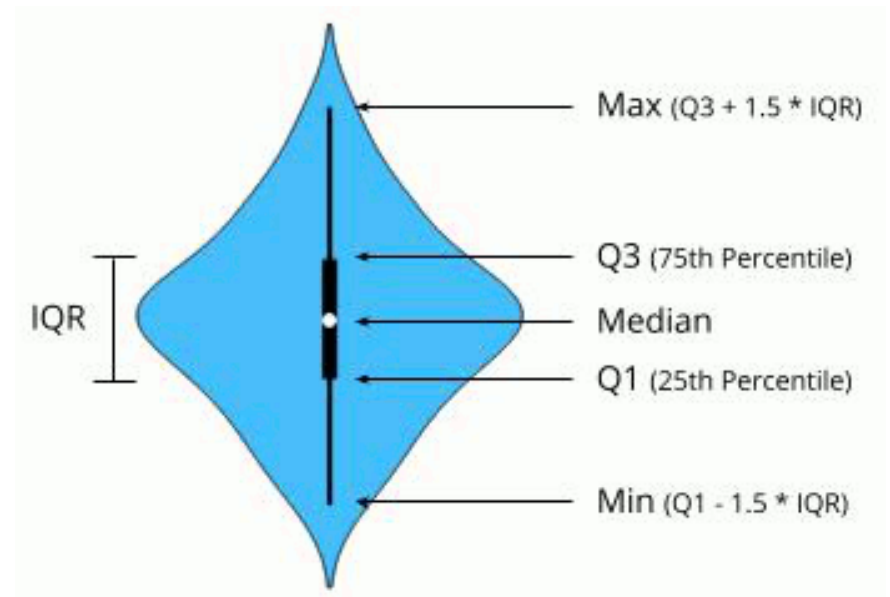


Box plot



Grafički prikaz rezultata

- Violin plot – dalje bolji uvid u distribuciju rješenja
- Prikazuje iste stvari kao box plot, ali još ilustrira i gustoću rješenja



Koji je algoritam bolji?

- Možemo li reći da je bolji onaj algoritam koji ima manji medijan ili srednju vrijednost?

Koji je algoritam bolji?

	Algoritam 1	Algoritam 2
	1.1	1
	2.2	2
	2.9	3
	3.7	4
	5.1	5
	6.2	6
	7.1	7
Srednja vrijednost	4.04	4
Medijan	3.7	4

Statistički testovi

- Samo na temelju srednje vrijednosti (ili medijana) nije moguće odrediti koji je algoritam bolji/lošiji
- Kako su algoritmi stohastički, moguće je da smo samo „nesrećom” dobili drugačije rezultate
- Potreban je alat koji će garantirati da postoji razlika između rezultata (uz zadanu vjerojatnost da takav zaključak nije pogrešan)

Statistički testovi

- Služe za usporedbu rezultata
 - Postoji li razlika između rezultata
 - Je li jedan rezultat bolji od drugoga
- Kako možemo to odrediti
 - Trebali bi znati cijelu populaciju (svo mogući rezultati algoritma)-> nije moguće

Statistički testovi

- Nul hipoteza – dva rezultata su jednaka
- Alternativna hipoteza:
 - Rezultati su različiti
 - Jedan rezultat je bolji od drugoga
- Rezultat nije 100% siguran -> za to bi morali znati cijelu populaciju rješenja
- Odabiremo razinu značajnosti testa *p-vrijednost*
 - Koliko je vjerojatno da smo dobili ovaj rezultat a da je nul hipoteza točna
 - Obično se koristi razina značajnosti 0.05

Statistički testovi

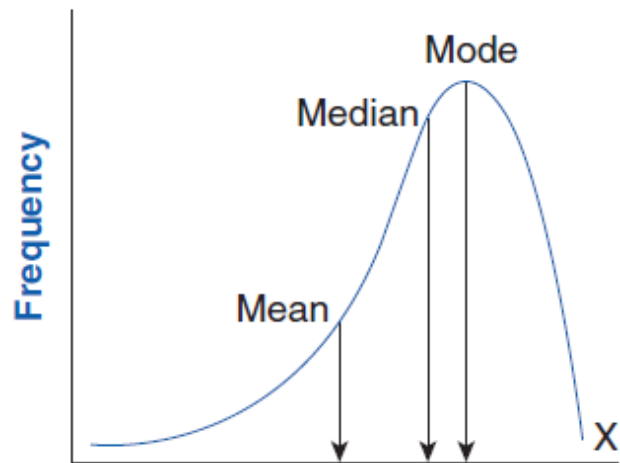
- Parametarski
 - Imaju određene pretpostavke od podacima
 - Nezavisnost, normalna distribucija...
 - Jači ako su pretpostavke zadovoljene
- Neparametarski
 - Ne koriste nikakve pretpostavke o podacima
 - Koristiti ako pretpostavke za neparametarske testove nisu zadovoljene

T-test

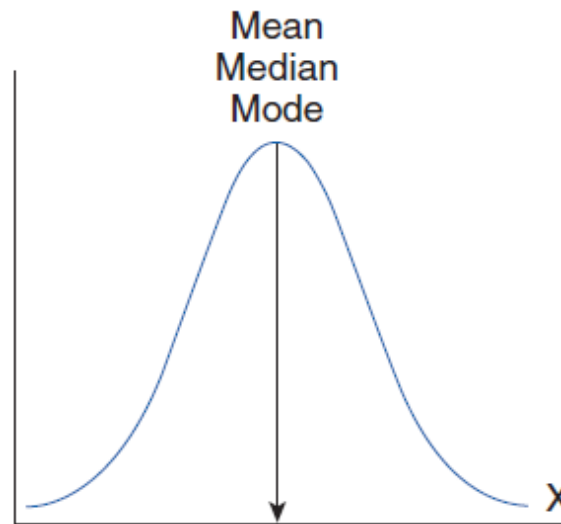
- Parametarski test
- Podaci moraju biti normalno distribuirani
- Procjenjuje postoji li značajna razlika između srednjih vrijednosti dviju grupa

Oblik distribucije

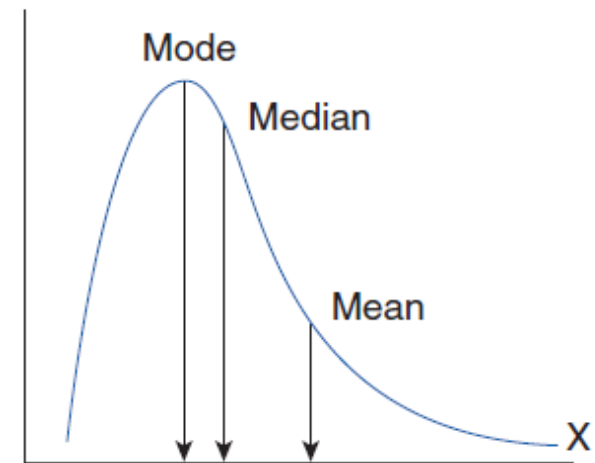
(a) Negatively skewed



(b) Normal (no skew)



(c) Positively skewed



Mann-Whitney test

- Neparametarski test
- Koristiti kada podaci ne prate normalnu distribuciju
 - Možemo provjeriti Shapiro-Wilk testom
- Uspoređuje medijane i određuje postoji li razlika između medijana dvije populacije

Usporedba više eksperimenata

- Eksperiment: pokretanje jednog algoritma na jednom problemu, zadani broj puta
- Želimo odrediti postoji li razlika između njih
- Kako?
- Odaberemo svaku kombinaciju algoritama i izračunamo postoji li statistička razlika
- Što će se desiti?

Usporedba više eksperimenata

- Ako radimo više individualnih usporedbi, raste nam i moguća greška koju smo napravili
- Više neće biti 5% nego veća
- Zbog toga postoje posebni testovi za usporedbu više algoritama

Usporedba više eksperimenata

- Test određuje postoji li značajna razlika između više grupa eksperimenata
- ANOVA
 - Parametarski
- Kruskal-Wallis
 - Neparametarski

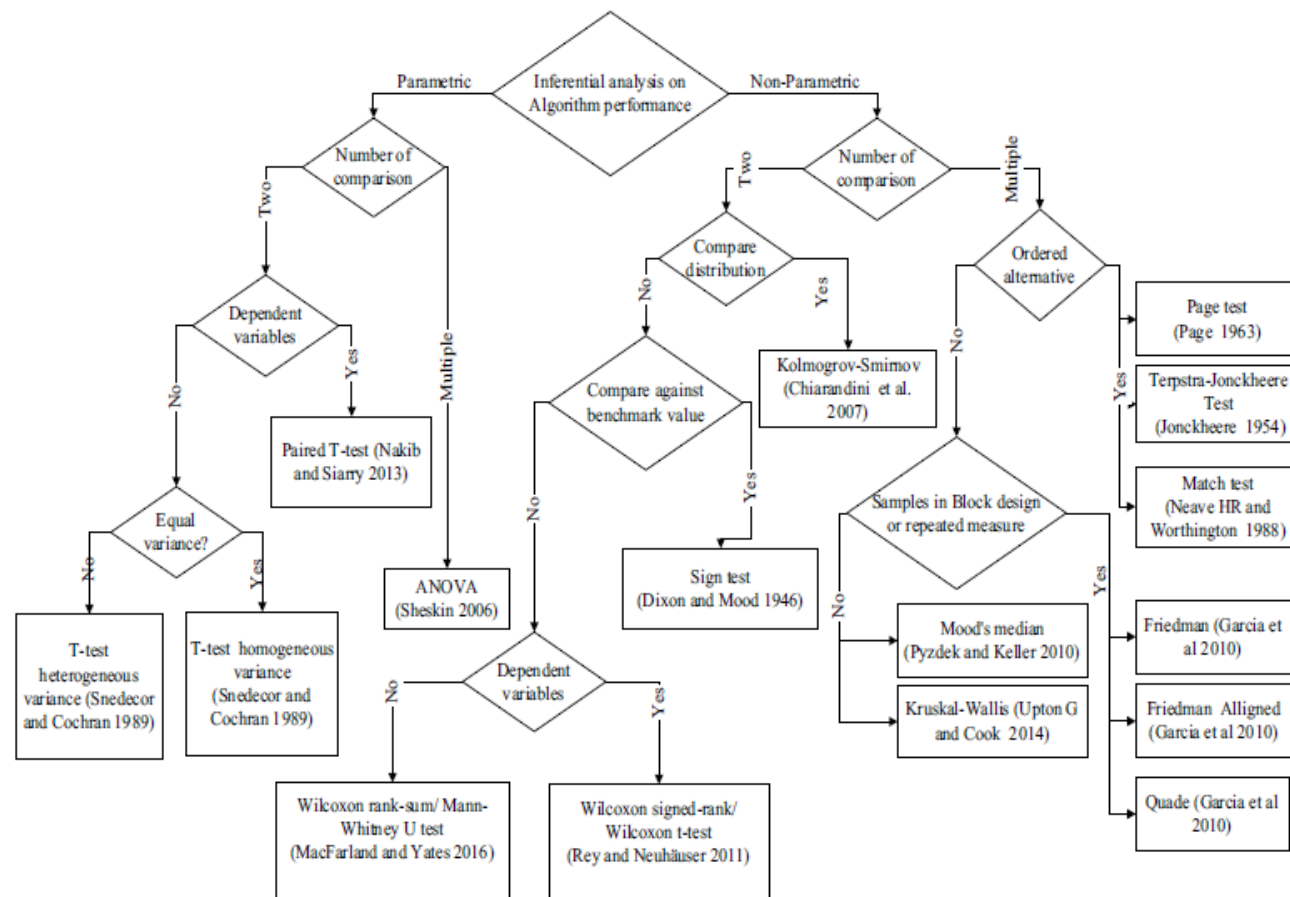
Usporedba više eksperimenata

- Odgovaraju samo na pitanje postoji li razlika ili ne
- Ako ne postoji, ne trebamo više informacija
- Ako postoji željeli bi znati kakva razlika postoji
 - Kako?
 - Post-hoc analizom

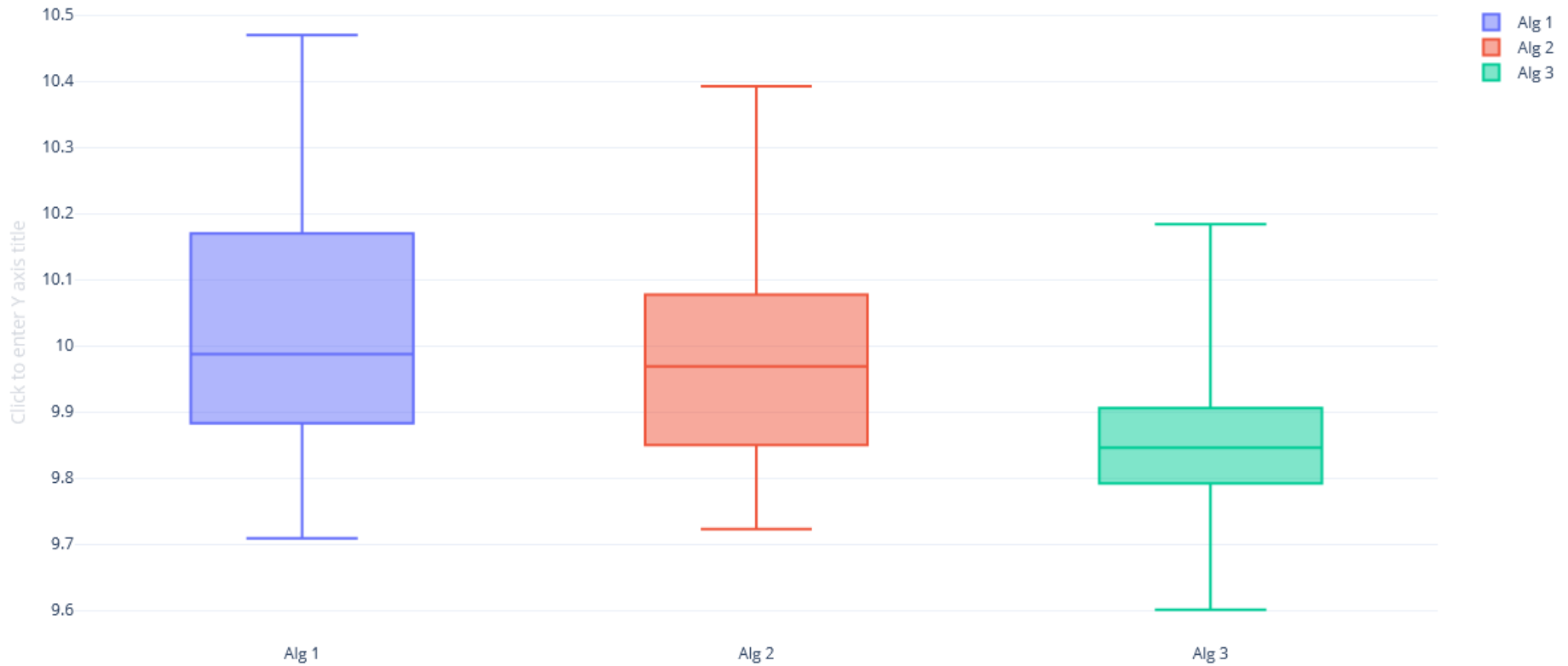
Usporedba više eksperimenata

- Test koji se provodi nakon kako bi se odredile postoje li razlike između parova eksperimenata
- Osiguravaju da je značajnost svake usporedbe jednaka sa zadanom p-vrijednosti
- Npr. Bonferroni-Dunn, Hochberg, ...

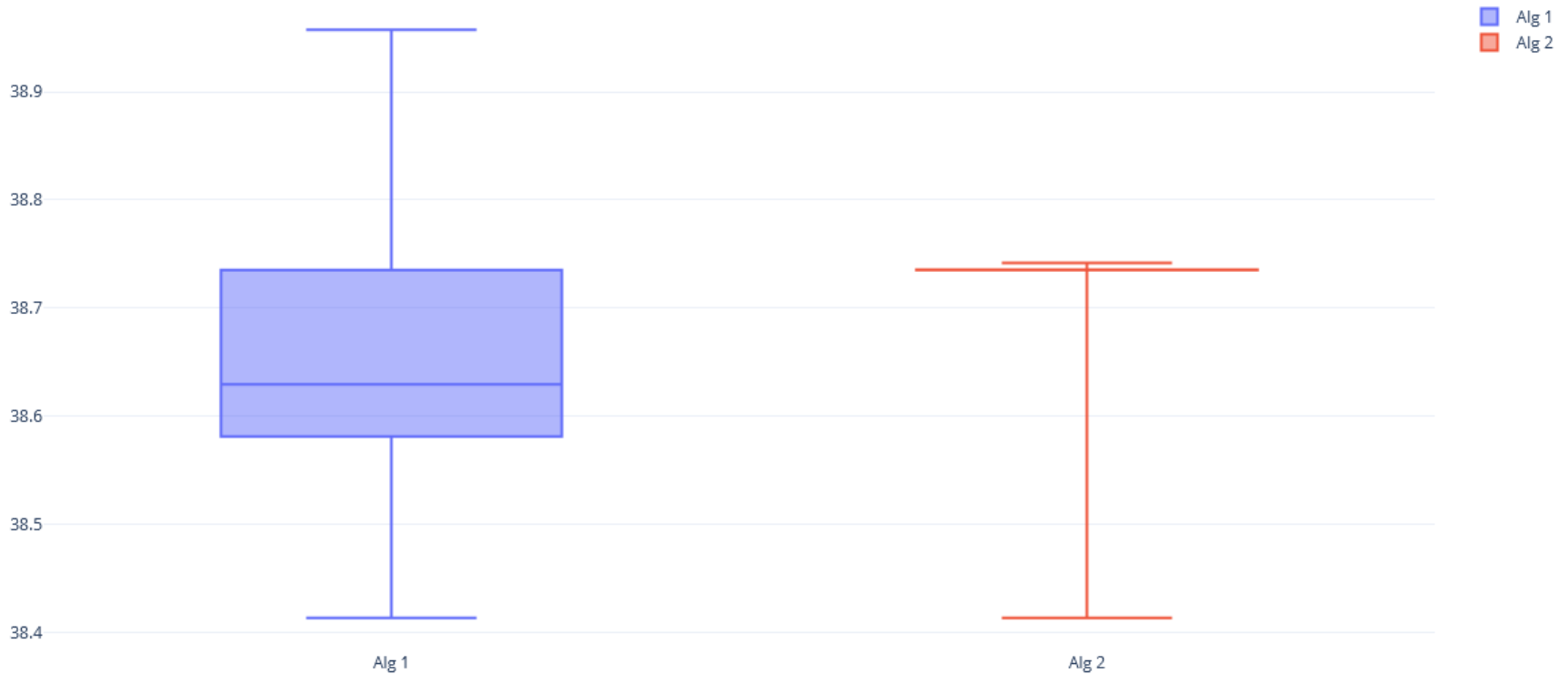
Pregled statističkih testova



Usporedba algoritama



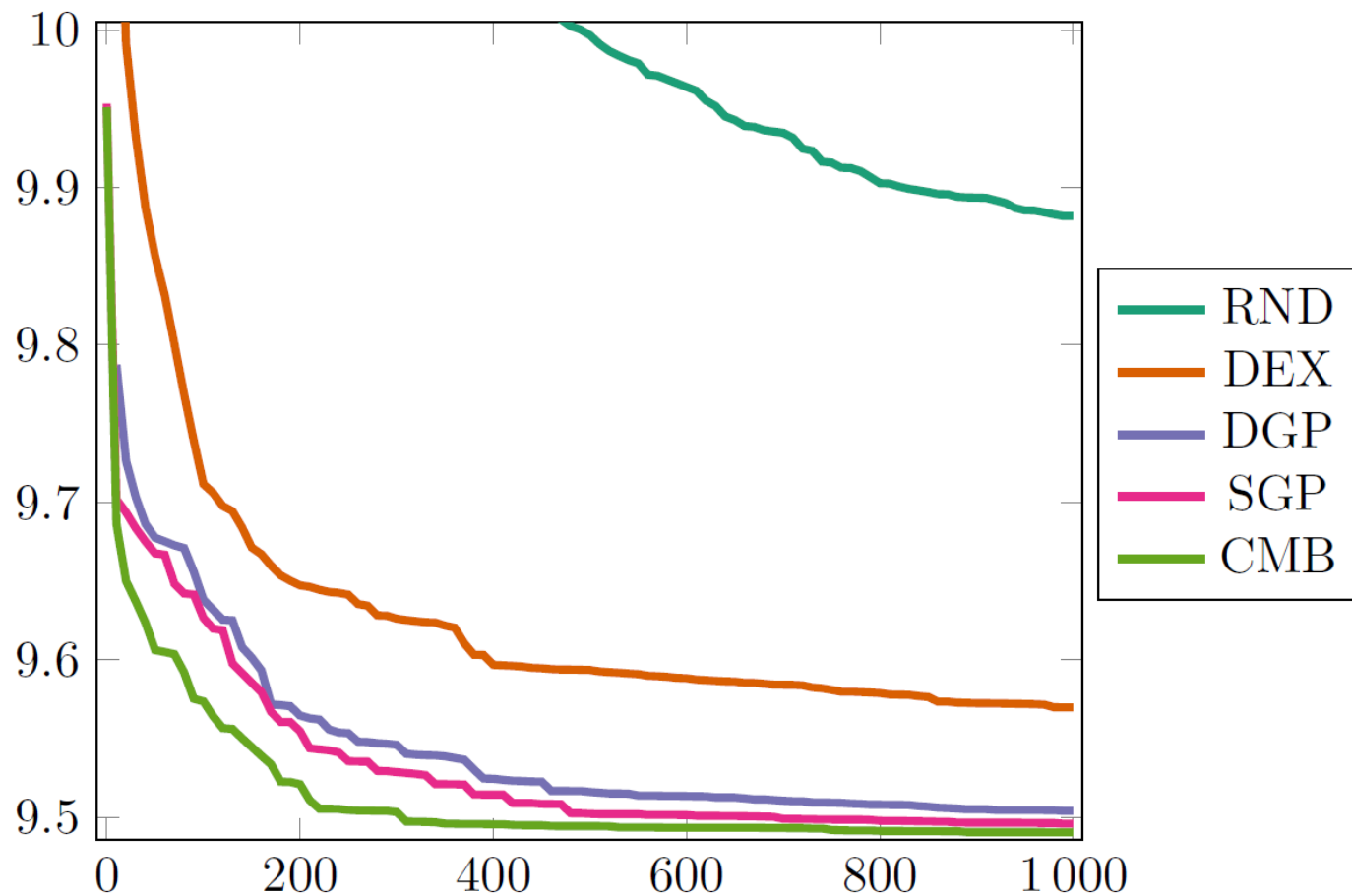
Usporedba algoritama



Konvergencija

- Koliko brzo algoritam dođe do dobrih rješenja
- Može poslužiti za usporedbu algoritama ako dostignu slična rješenja

Konvergenција



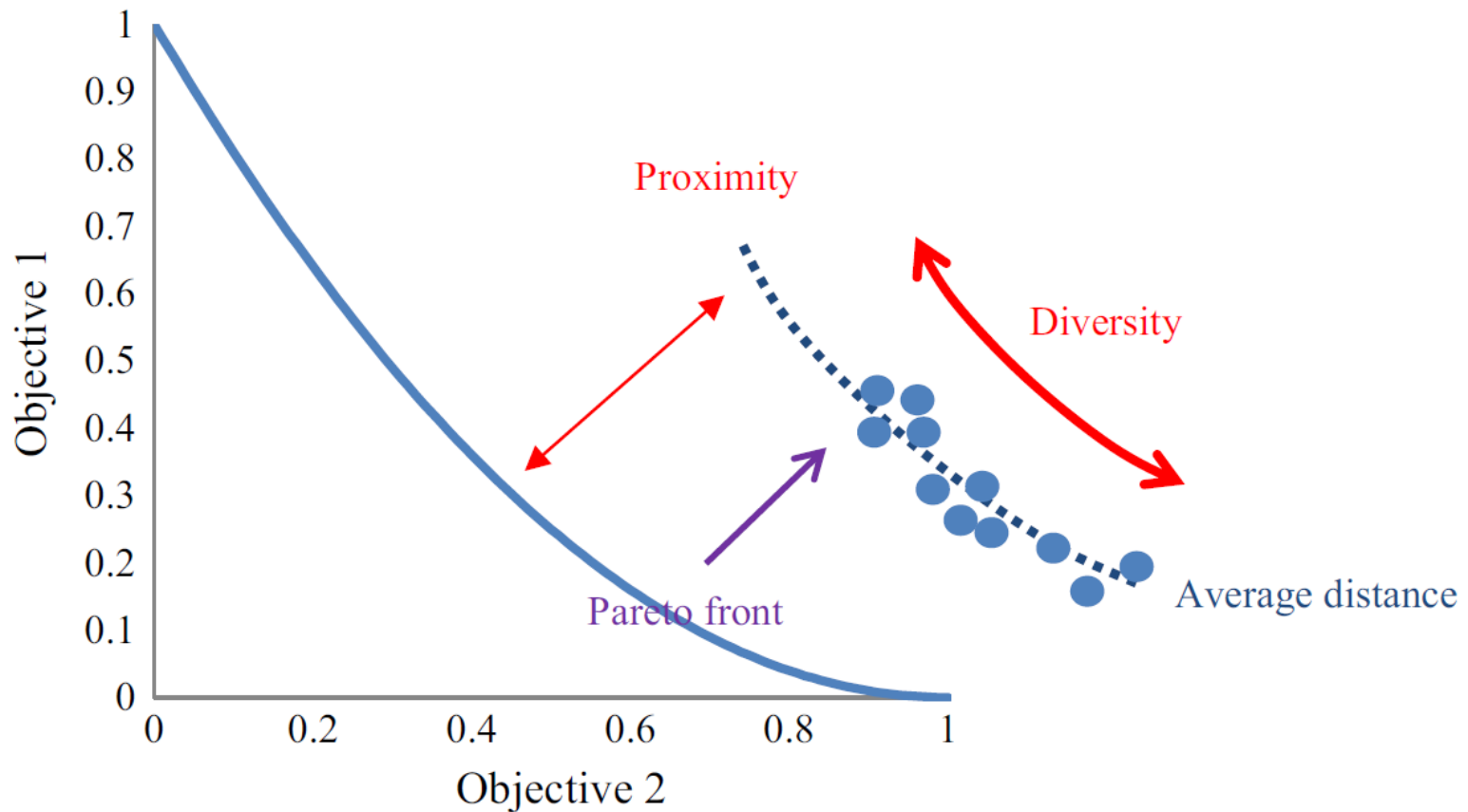
Usporedba višekriterijskih algoritama

- Kako uspoređivati skupove rješenja?
- Ne možemo izračunati standardne statističke metrike

Usporedba višekriterijskih algoritama

- Želimo vidjeti koliko je naš skup rješenja dobar
- Imamo različita svojstva koja želimo mjeriti:
 - Kapacitet
 - Konvergencija
 - Diverzifikacija

Usporedba višekriterijskih algoritama



Kapacitet

- Mjeri količinu nedominiranih rješenja u skupu dobivenih rješenja
- Npr. broj nedominiranih rješenja ili omjer nedominiranih rješenja u skupu
- Obično želimo maksimizirati ovu mjeru
- Rijetko se koristi za usporedbu

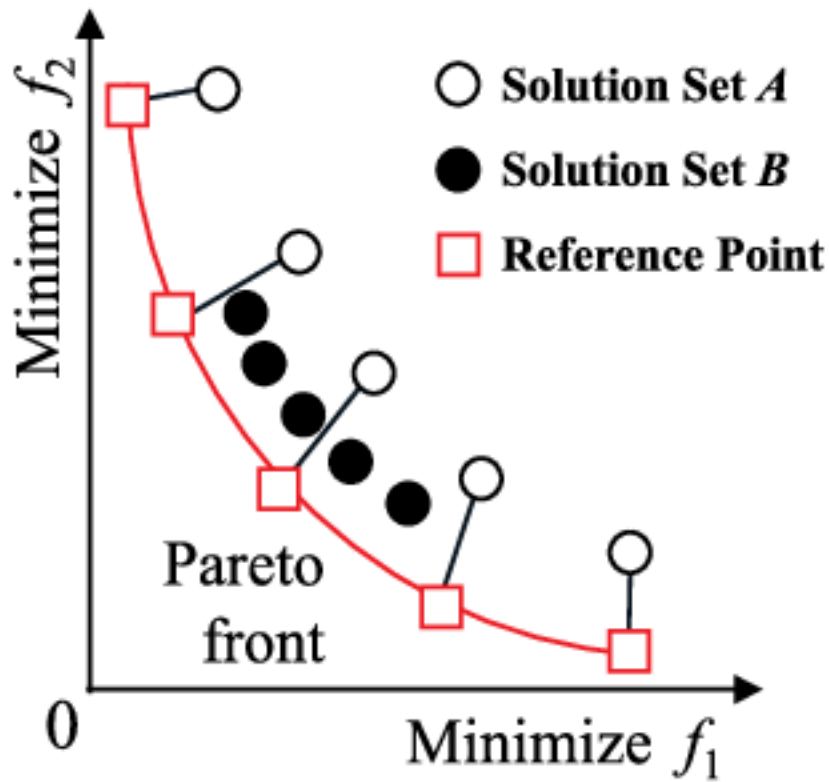
Konvergencija

- Mjere koliko je dobivena Pareto fronta (S) blizu pravoj Pareto fronti (P)
- *Generational distance* (GD):

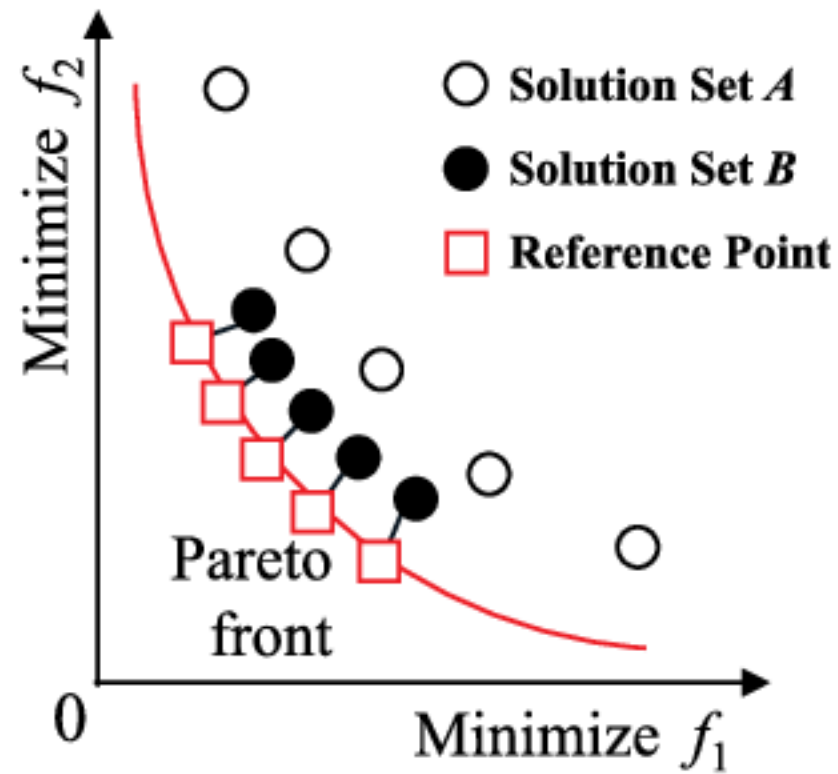
$$GD(S, P) = \frac{\left(\sum_{i=1}^{|S|} d_i^q \right)^{\frac{1}{q}}}{|S|}$$

- d_i najmanja udaljenost rješenja iz S ka najbližoj točki iz P
- $q=2$

Generational distance



(a)



(b)

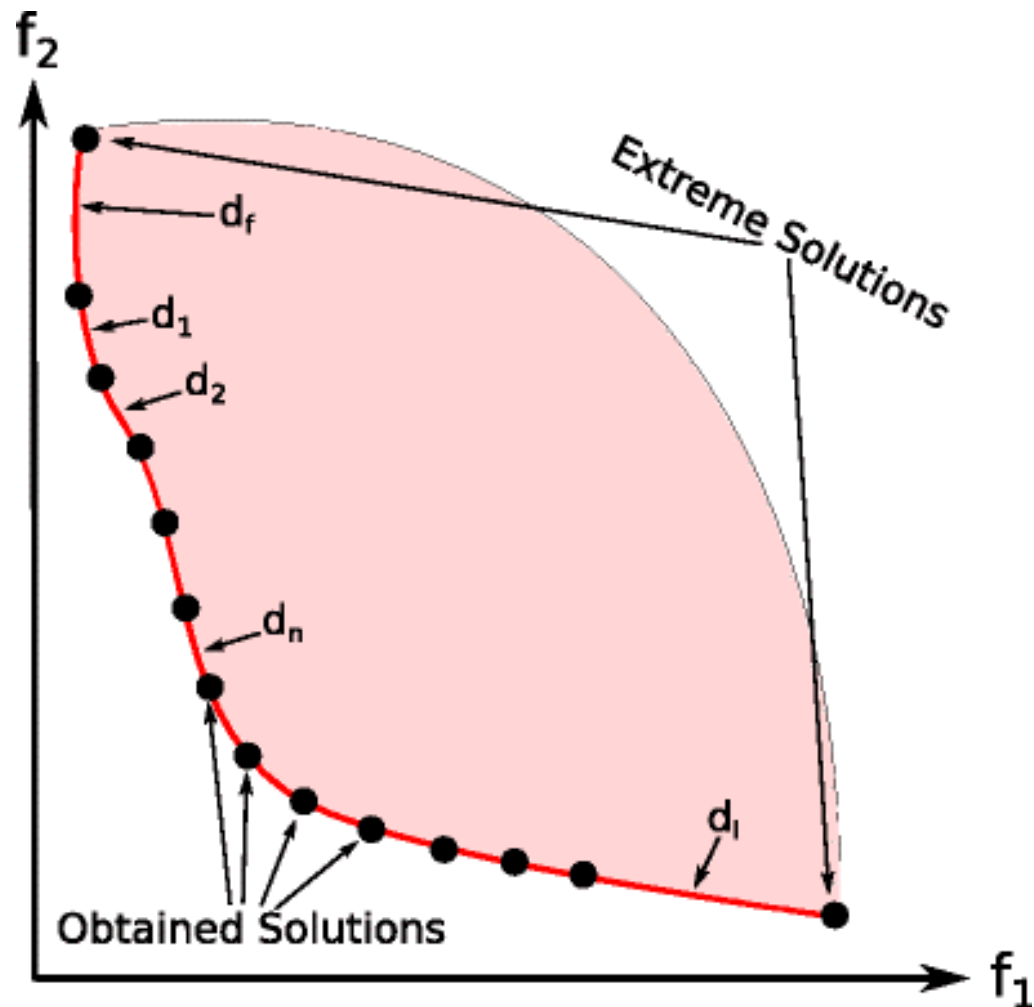
Diverzifikacija

- Mjeri koliko su rješenja raznolika, tj. koliko dobro pokrivaju prostor
- SPREAD

$$SPREAD = \frac{d_f + d_l + \sum_{i=1}^{|S|-1} |d_i - \bar{d}|}{d_f + d_l + (|S| - 1)\bar{d}}$$

- d_f, d_l najmanje Euklidske udaljenosti rubnih rješenja
- d_i udaljenost uzastopnih rješenja
- \bar{d} srednja vrijednost udaljenosti

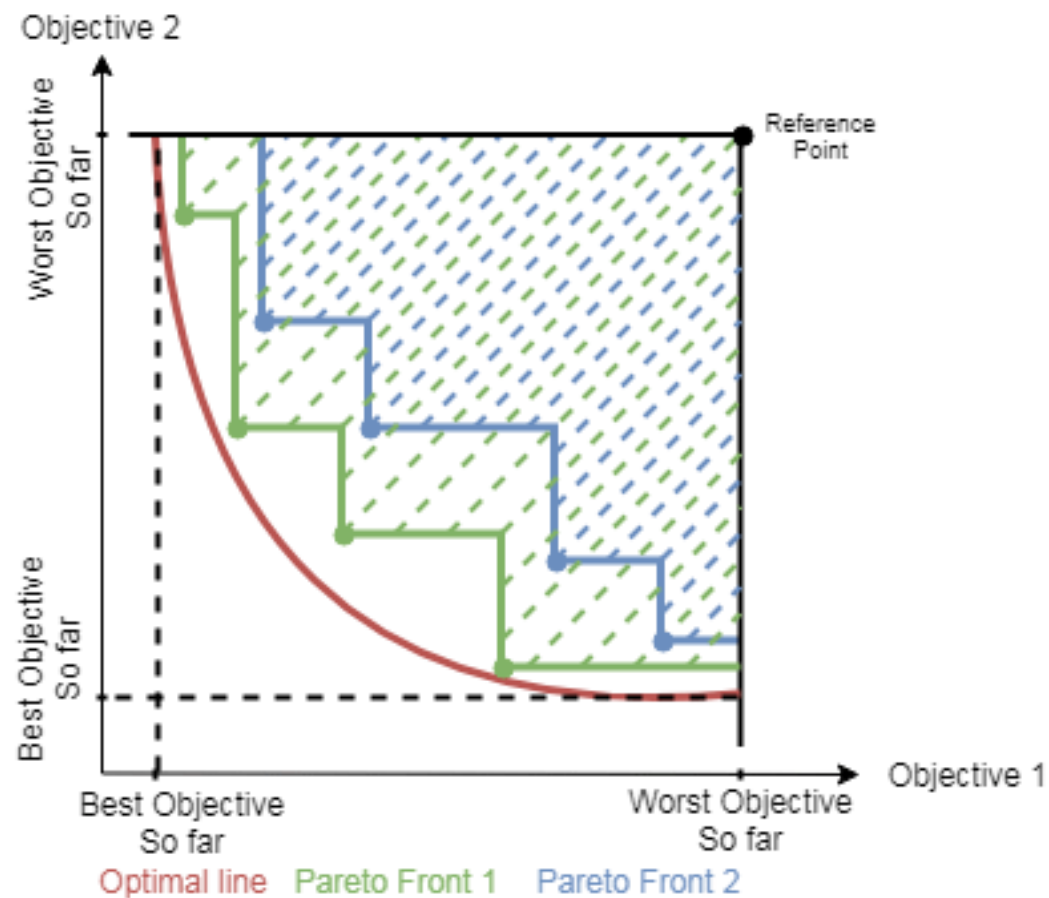
SPREAD



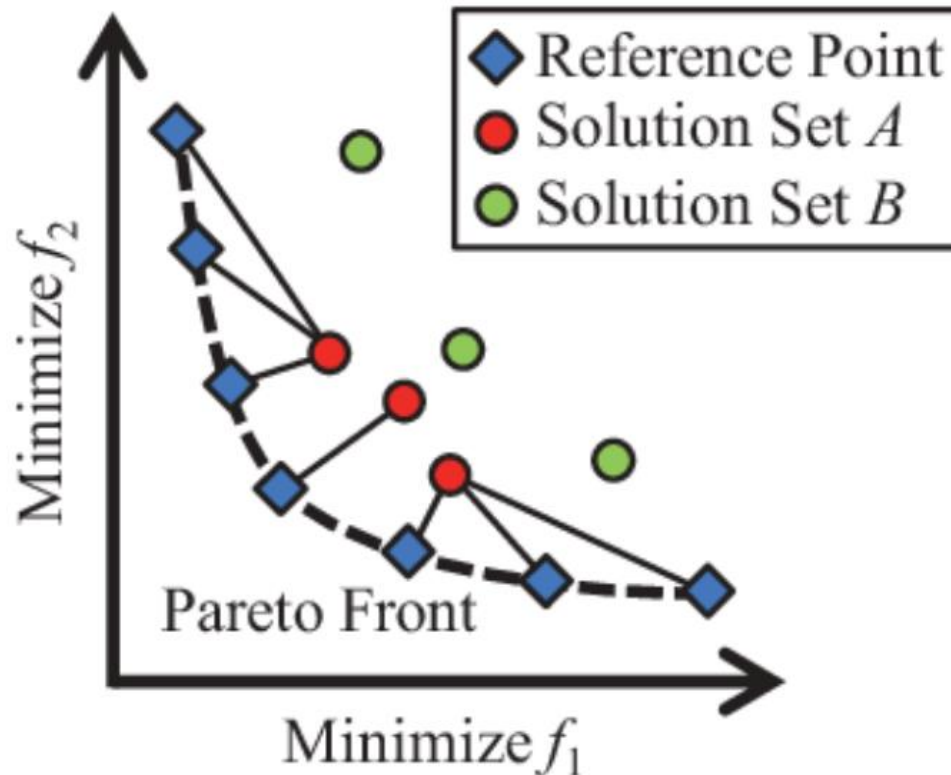
Konvergencija i diverzifikacija

- Prethodne mjere opisuju samo jednu karakteristiku
- Bolje bi bilo imati mjere koje opisuju i konvergenciju i diverzifikaciju rješenja
- Hipervolumen
- Inverted generational distance

Hipervolumen



Inverted generational distance



Podešavanje parametara

- Algoritmi uvelike ovise o odabranim parametrima
- Potrebno je odabrati prikladne parametre za algoritme
- Parametri ovise o algoritmu:
 - Veličina populacije
 - Vjerojatnost mutacije
 - Početna temperatura
 - ...

Podešavanje parametara

- Grid search
 - Za svaki parametar odabrati određene vrijednosti
 - Isprobati sve kombinacije
 - Moguće samo za jako lagane probleme
 - Kombinatorna eksplozija

Podešavanje parametara

- Optimizirati parametar po parametar
- Za svaki parametar odabrati neku „dobru” vrijednost
- Varirati vrijednosti za jedan parametar i odabrati najbolju vrijednost i fiksirati ju
- Nastaviti za ostale parametre
- Osigurati da algoritmi odrade istu količinu posla
- Jednostavna i često korištena metoda

Podešavanje parametara

- Metode za automatsko podešavanje parametara
 - Irace
 - ParamILS
 - I mnoge druge

Koji algoritam koristiti?

- Postoji mnoštvo standardnih evolucijskih algoritama
- Postoje mnogi hibridni algoritmi, kombinacije različitih metoda (npr. genetski algoritam i simulirano kaljenje)
- Svako malo predlažu se nove vrste evolucijskih algoritama

No Free Lunch

- Ne postoji algoritam koji je najbolji za apsolutno sve probleme ili instance problema
- Moguće je pronaći probleme za koje će neki algoritam biti bolji od nekog drugog i obrnuto

Koji algoritam koristiti?

<u>Simulated Annealing</u>	SA	Trajectory-based	-	1983
<u>Tabu Search</u>	TS	Trajectory-based	-	1989
<u>Genetic Algorithm</u>	GA	Evolutionary-based	-	1992
<u>Evolutionary Algorithm</u>	EA	Evolutionary-based	-	1994
<u>Cultural Algorithm</u>	CA			1994
<u>Particle Swarm Optimization</u>	PSO	Nature-inspired	Swarm-based	1995
Variable Neighborhood Search	VNS	Trajectory-based	-	1997
<u>Local Search</u>	LS			1997
Differential Evaluation	DE	Evolutionary-based	-	1997
Guided Local Search	GLS	Trajectory-based	-	1998
Clonal Selection Algorithm	CSA	Evolutionary-based	-	2000
<u>Harmony Search</u>	HS	Evolutionary-based	-	2001
<u>Memetic Algorithm</u>	MA	Evolutionary-based	-	2002
Iterative Local Search	ILS	Trajectory-based	-	2003

Koji algoritam koristiti?

Artificial Bee Colony	ABC	Nature-inspired	Bio-inspired	2005
Shuffled Frog Leaping Algorithm	SFLA	Nature-inspired	Bio-inspired	2006
Invasive Weed Optimization	IWO	Nature-inspired	Plant-based	2006
<u>Glowworm Swarm Optimization</u>	GSO	Nature-inspired	Swarm-based	2006
<u>Ant Colony Optimization</u>	ACO	Nature-inspired	Bio-inspired	2006
Imperialistic Competitive Algorithm	ICA	Nature-inspired	Human-based	2007
Monkey Algorithm	MA	Nature-inspired	Bio-inspired	2008
Intelligent Water Drops	IWD	Nature-inspired	Swarm-based	2008
Firefly Algorithm	FA	Nature-inspired	Bio-inspired	2008
Biogeography Based Optimization	BBO	Nature-inspired	Human-based	2008
Key Cutting Algorithm	KCA			2009
Hunting Search	HS	Nature-inspired	Swarm-based	2009
Group Search Optimizer	GSO	Nature-inspired	Swarm-based	2009
Cuckoo Search	CS	Nature-inspired	Bio-inspired	2009
Chemical Reaction Optimization	CRO	Nature-inspired	Physics/Chemistry-based	2009

Koji algoritam koristiti?

Fireworks Algorithm	FWA			2010
Eagle Strategy	ES	Nature-inspired		2010
Charged System Search	CSS	Nature-inspired	Physics/Chemistry-based	2010
Bat Algorithm	BA	Nature-inspired	Bio-inspired	2010
Teaching-Learning-Based Optimization	TLBO			2011
Stochastic Diffusion Search	SDS			2011
Cuckoo Optimization Algorithm	COA	Nature-inspired	Bio-inspired	2011
Water Cycle Algorithm	WCA			2012
Migrating Birds Optimization	MBO	Nature-inspired	Swarm-based	2012
Krill Herd Algorithm	KHA	Nature-inspired	Bio-inspired	2012
Fruit Fly Optimization	FFO			2012
Bacterial Colony Optimization	BCO			2012
Dolphin Echolocation	DE	Nature-inspired	Bio-inspired	2013
Black Hole Algorithm	BH	Nature-inspired	Physics/Chemistry-based	2013
Backtracking Search Algorithm	BSA	Evolutionary-based	-	2013
Animal Migration Optimization	AMO	Nature-inspired	Swarm-based	2013

Koji algoritam koristiti?

Spider Monkey Optimization	SMO	Nature-inspired	Bio-inspired	2014
Seeker Optimization Algorithm	SOA			2014
SDA Optimization Algorithm	SDA	Nature-inspired	Bio-inspired	2014
Radial Movement Optimization	RMO			2014
Keshtel Algorithm	KA	Nature-inspired		2014
Flower Pollination Algorithm	FPA	Nature-inspired	Plant-based	2014
Coral Reefs Optimization Algorithm	CROA			2014
Colliding Bodies Optimization	CBO			2014
Chicken Swarm Optimization	CSO			2014
Bumble Bees Mating Optimization	BBMO			2014
Artificial Root Foraging Algorithm	ARFA	Nature-inspired	Plant-based	2014
Water Wave Optimization	WWA	Nature-inspired	Physics/Chemistry-based	2015
Vortex Search Algorithm	VSA	Nature-inspired	Physics/Chemistry-based	2015
Stochastic Fractal Search	SFF	Evolutionary-based	-	2015

Koji algoritam koristiti?

Moth-Flame Optimization	MFO	Nature-inspired	Bio-inspired	2015
Locust Swarm Algorithm	LSA	Nature-inspired	Swarm-based	2015
Lightning Search Algorithm	LSA	Nature-inspired	Physics/Chemistry -based	2015
Jaguar Algorithm	JA	Nature-inspired	Bio-inspired	2015
Grey Wolf Optimizer	GWO	Nature-inspired	Bio-inspired	2015
Elephant Search Algorithm	ESA	Nature-inspired	Bio-inspired	2015
Cricket Algorithm	CA			2015
Central Force Optimization	CFO			2015
Bottlenose Dolphin Optimization	BDO	Nature-inspired		2015
Artificial Fish Swarm Algorithm	AFSA	Nature-inspired		2015
Artificial Algae Algorithm	AAA			2015
Ant Lion Optimizer	ALO	Nature-inspired	Bio-inspired	2015
Alienated Ant Algorithm	AAA			2015
African Buffalo Optimization	ABO	Nature-inspired	Swarm-based	2015
Adaptive Dimensional Search	ADS			2015

Koji algoritam koristiti?

Whale Optimization Algorithm	WOA	Nature-inspired	Bio-inspired	2016
Virus Colony Search	VCS	Nature-inspired	Bio-inspired	2016
Sine Cosine Algorithm	SCA	Nature-inspired	Physics/Chemistry-based	2016
Red Deer Algorithm	RDA	Nature-inspired	Bio-inspired	2016
Lion Optimization Algorithm	LOA	Nature-inspired	Bio-inspired	2016
Joint Operations Algorithm	JOA	Nature-inspired	Swarm-based	2016
Electromagnetic Field Optimization	EFO	Nature-inspired	Physics/Chemistry-based	2016
Crow Search Algorithm	CSA	Nature-inspired	Bio-inspired	2016
Across Neighborhood Search	ANS	Evolutionary-based	-	2016
Squirrel Search Algorithm	SSA	Nature-inspired	Bio-inspired	2018
Social Engineering Optimizer	SEO	Nature-inspired	Human-based	2018
Phototropic Optimization Algorithm	POA	Nature-inspired	Plant-based	2018
Owl Search Algorithm	OSA	Nature-inspired	Bio-inspired	2018
Coyote Optimization Algorithm	COA	Nature-inspired	Swarm-based	2018

Koji algoritam koristiti?

Thermal Exchange Optimization	TEO	Nature-inspired	Physics/Chemistry-based	2019
Harris Hawks Optimization	HHO	Nature-inspired	Bio-inspired	2019
Future Search Algorithm	FSA	Nature-inspired	Human-based	2019
Emperor Penguins Colony	EPC	Nature-inspired	Swarm-based	2019
Stochastic Paint Optimizer	SPO	Art-inspired	Color-based	2020
Political Optimizer	PO	Nature-inspired	Human-based	2020
Heap-Based Optimizer	HBO	Nature-inspired	Human-based	2020
Giza Pyramids Construction	GPC	Ancient-inspired	-	2020
Color Harmony Algorithm	CHA	Art-inspired	Color-based	2020

Koji algoritam koristiti?

- Većina algoritama radi na sličan način, samo koriste drugačiju metaforu
- Koristiti standardne algoritme, po mogućnosti različite vrste
- Koristiti dokazano dobre algoritme, npr. CMA-ES
- Ako jednostavnim varijantama algoritma nije moguće dobiti zadovoljavajuće rješenje, onda prilagoditi algoritam problemu (bolje nego isprobati 100 algoritama)

Kako odabrati prikladan algoritam?

- Odabrati skup problema za testiranje
- Optimizirati parametre
- Napraviti statističku analizu rezultata

Na čemu testirati algoritam

- Standardni benchmark problemi:
 - TSP: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>
 - Kontinuirana opt: <https://www.sfu.ca/~ssurjano/optimization.html>
 - Višekriterijska opt: <https://link.springer.com/article/10.1007/s40747-017-0039-7> i https://en.wikipedia.org/wiki/Test_functions_for_optimization#Test_functions_for_multi-objective_optimization
 - Simbolička regresija: https://dev.heuristiclab.com/trac.fcgi/blog/gkronber/symbolic_regression_benchmark
 - Itd...

Podjela skupa instanci problema

- Ovisi o problemu
- Kod klasične optimizacije – bez podjele ili se optimizacija parametara provodi na nekom podskupu
- Kod problema strojnog učenja
 - Skup za učenje -> na njemu se rješenja evaluiraju tijekom izvođenja evolucijskih algoritama
 - Skup za validaciju -> na njemu optimiramo parametre
 - Skup za testiranje -> na njemu se provodi konačna ocjena algoritama