

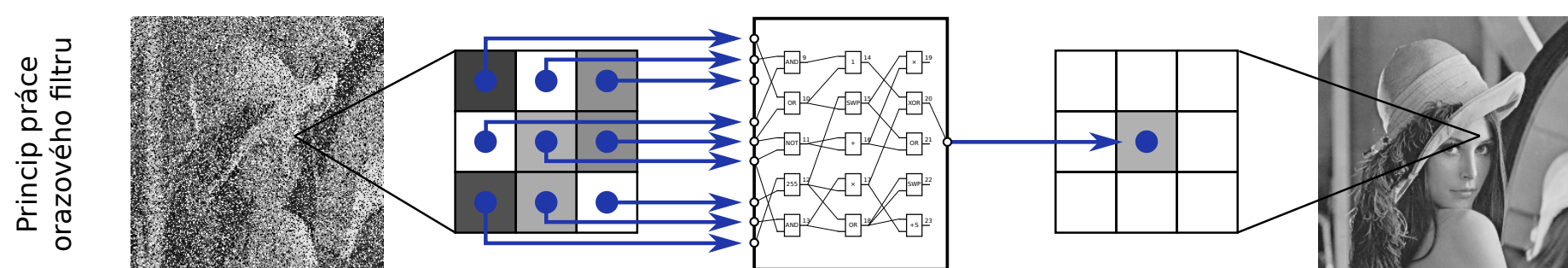
Souběžné učení v koevolučních algoritmech

Michal Wiglasz (xwigla00@stud.fit.vutbr.cz)

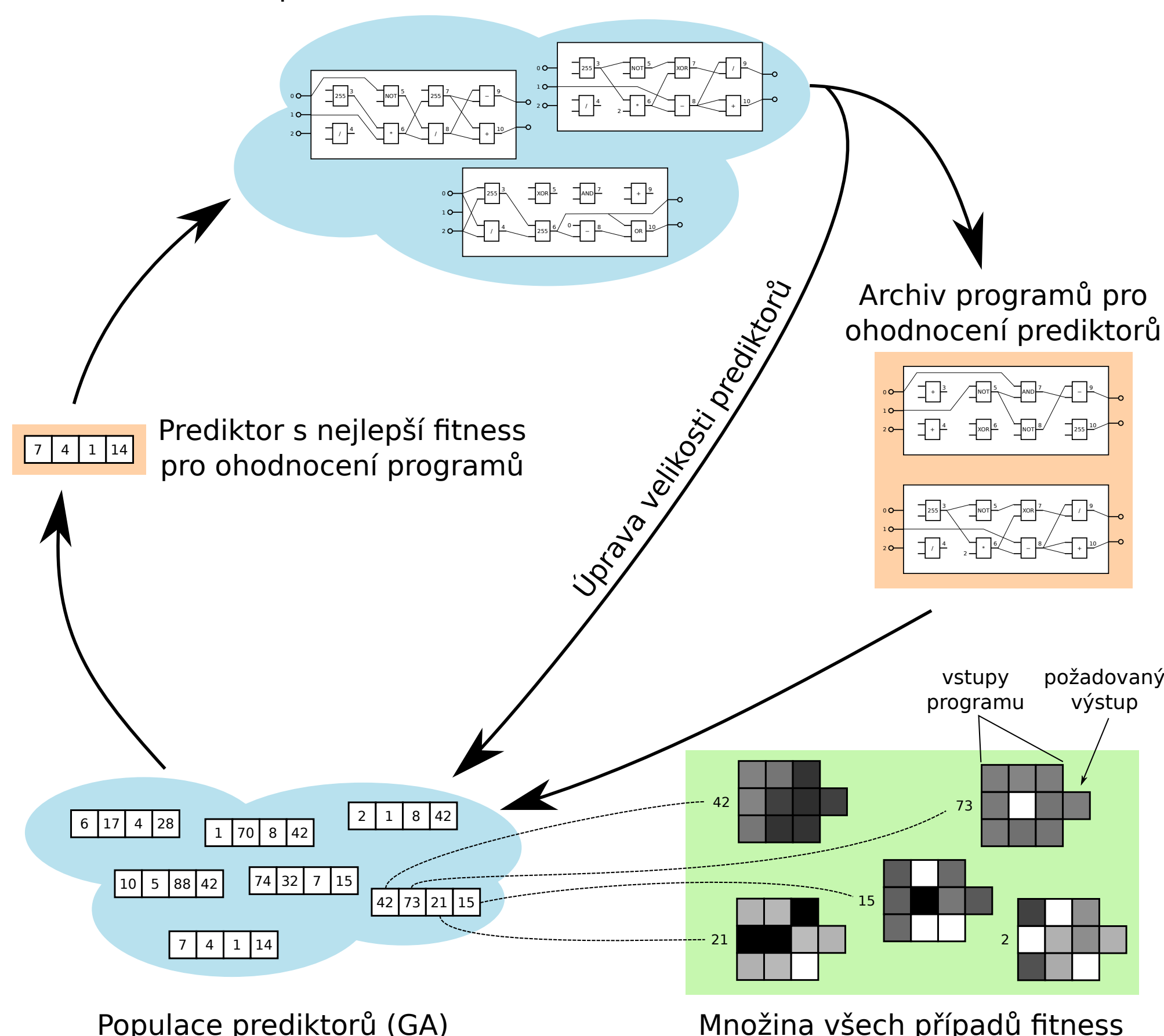
Příspěvek 27

Obrazové filtry

Většina obrazových filtrů pracuje s lokálním okolím pixelů, například devítiokolí. Jeho vstupem jsou hodnoty všech pixelů v okolí a výstupem je nová hodnota pixelu v jeho středu.

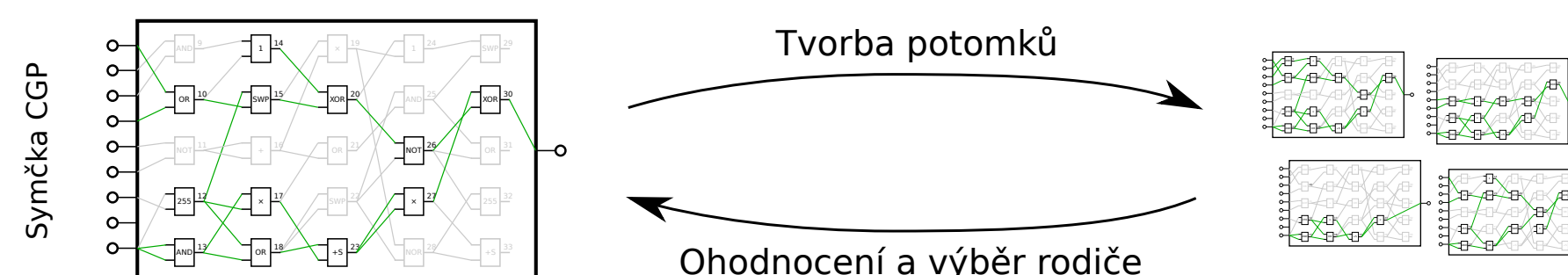


Populace kandidátních filtrů (CGP)



Kartézské genetické programování (CGP)

Obrazové filtry lze automatizovaně navrhovat pomocí kartézského genetického programování (CGP). Filtre je reprezentován kartézskou mřížkou výpočetních bloků. Jejich funkce a vzájemné propojení jsou nalezeny evolučním algoritmem.

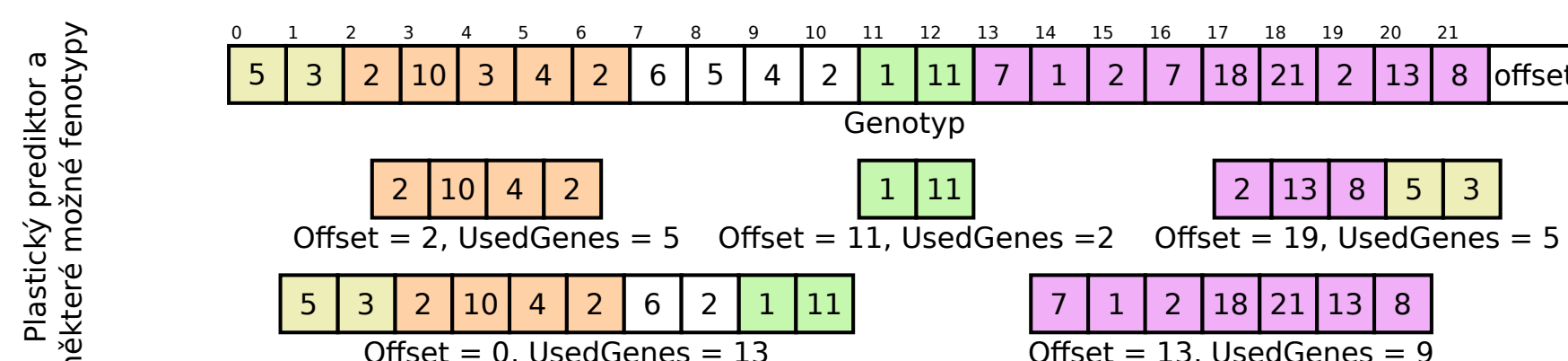


Prediktory fitness

Prediktory fitness slouží pro přibližné určení kvality filtru, čímž lze urychlit evoluci. Prediktor vybírá podmnožinu všech případů fitness, která se používá pro výpočet predikované fitness. Prediktory se vyvíjejí pomocí genetického algoritmu paralelně s obrazovými filtry.

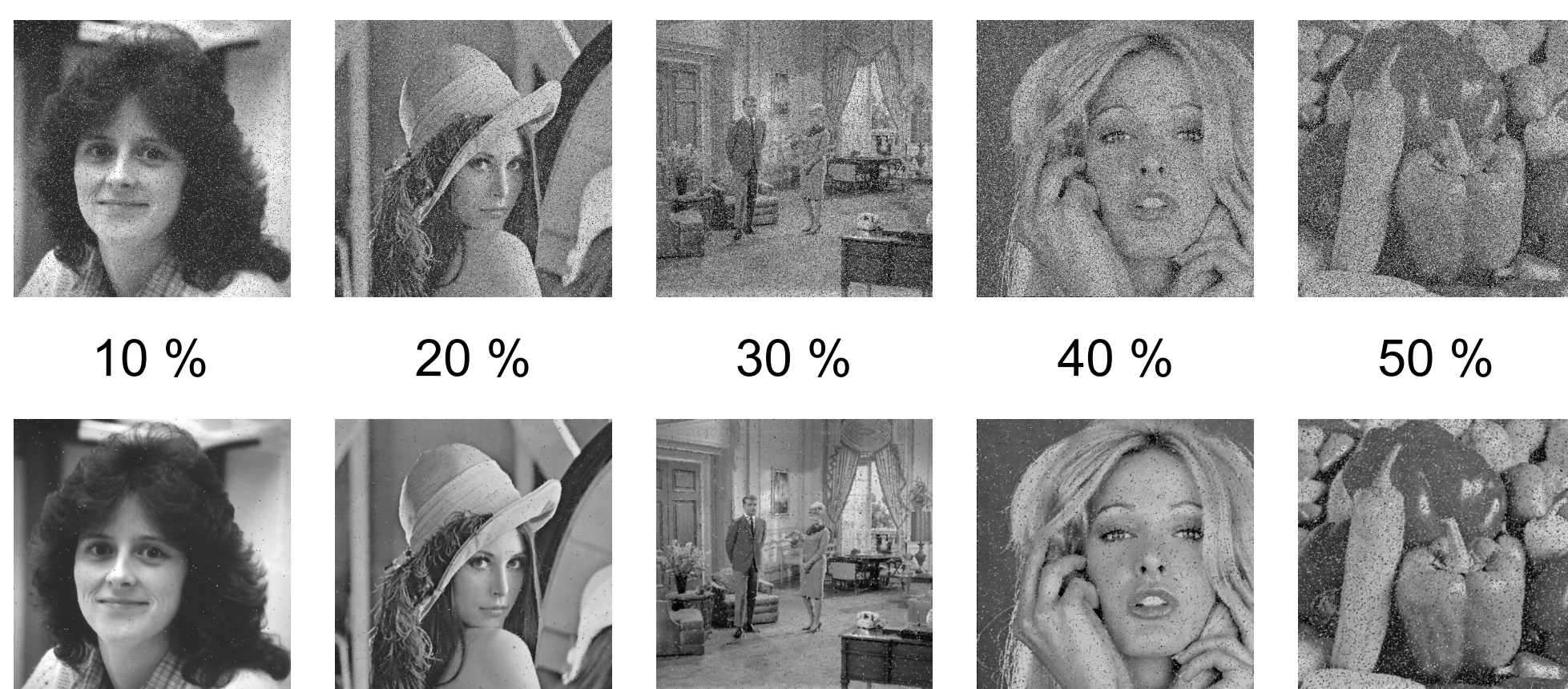
U souběžného učení mají prediktory plastický fenotyp, což znamená, že ze stejného genotypu mohou vzniknout různé fenotypy podle prostředí. Součástí prostředí je hodnota UsedGenes, která určuje, kolik genů se použije pro konstrukci fenotypu prediktoru, a tedy jeho velikost. Hodnota se upravuje podle vývoje v populaci filtrů nebo pokud se predikovaná fitness příliš liší od skutečné. Pravidla jsou tato:

- 1) Pokud je nepřesnost predikce příliš vysoká, prediktory se prodlouží.
- 2) Pokud se fitness nemění, evoluce pravděpodobně uvázla v lokálním optimu a prediktory se zkrátí, což může pomoci toto optima opustit.
- 3) Pokud fitness klesá, evoluce pravděpodobně opouští lokální optimum a mírné zkrácení prediktoru může pomoci postup urychlit.
- 4) Pokud fitness roste, prediktory se prodlouží a tím se zpřesní predikce.



Ukázka nalezených filtrů

Obrázky poškozené impulzním šumem

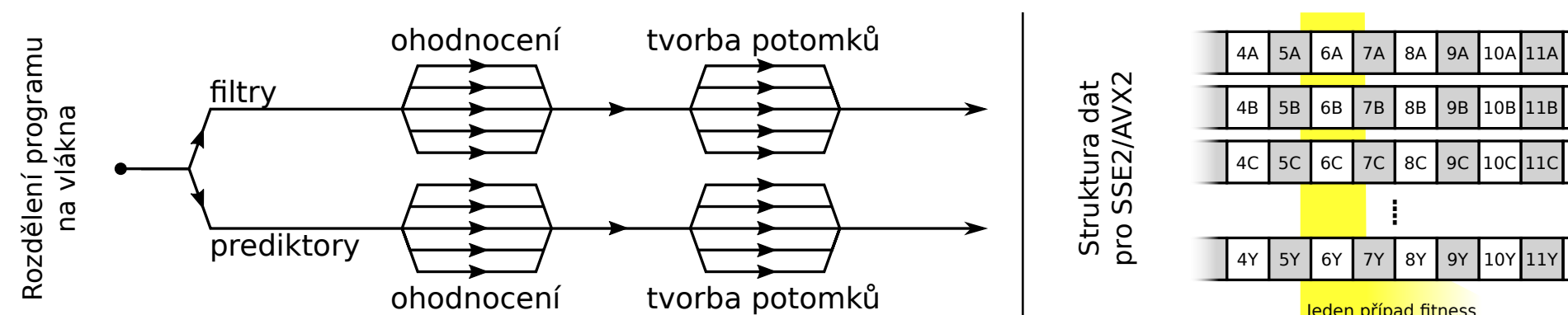


Výstup nejlepšího nalezeného filtru (1 krok filtrace)

Implementace algoritmu

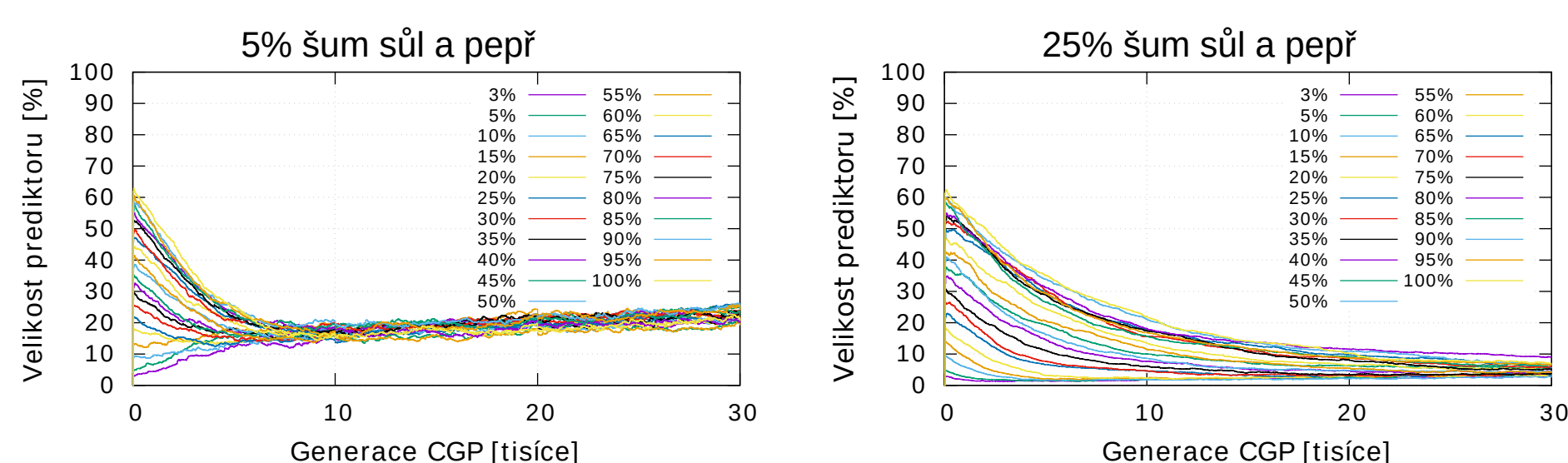
Program je implementován v jazyce C. Aby byl co nejrychlejší, je paralelizován pomocí OpenMP a výpočet výstupů filtrů je implementován také pomocí vektorových instrukcí SSE2 a AVX2. Pomocí nich lze zpracovávat 16 nebo 32 pixelů najednou. Implementace s SSE2 je přibližně 10× rychlejší, s AVX2 dokonce 16× (pro standardní CGP bez koevoluce).

Pro efektivní vektorové zpracování je třeba upravit i strukturu dat v paměti. Pro vektorové zpracování je výhodnější několik samostatných polí pro každou pozici v devítiokolí a požadovaný výstup. Také se musí připravit případy fitness určené prediktory tak, aby byly v paměti za sebou.



Schopnost adaptace velikosti prediktoru

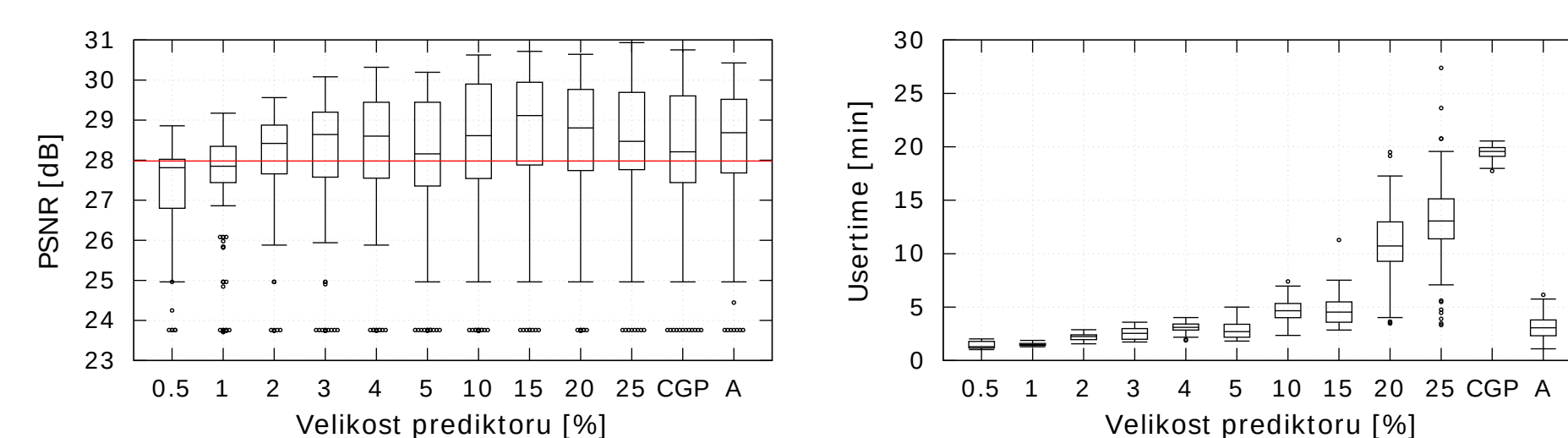
Bez ohledu na počáteční nastavení velikost prediktoru konverguje ke stejné hodnotě, která závisí na řešené úloze. Je ale výhodnější začít s kratšími prediktory, protože výpočet predikované fitness je pak rychlejší.



Vývoj velikosti prediktoru při různé počáteční velikosti

Srovnání s CGP a s koevolucí s pevnými prediktory

Kvalita získaných filtrů se příliš neliší od výsledků koevoluce s pevnými prediktory ani od výsledků standardního CGP. Oproti němu bylo dosaženo v průměru 8,39násobného zrychlení (součet procesorového času všech vláken). Také odpadá nutnost časově náročného experimentálního hledání nejvýhodnější velikosti prediktorů.



Srovnání kvality filtrů a potřebného času CGP a koevoluce CGP s pevnými i adaptivními (A) prediktory fitness pro 10% impulzní šum