



Universiteit
Leiden

The Netherlands

Sampling strategies in automated algorithm configuration

Anastacio, M.I.A.

Citation

Anastacio, M. I. A. (2026, June 23). *Sampling strategies in automated algorithm configuration*. Retrieved from <https://hdl.handle.net/1887/4307419>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4307419>

Note: To cite this publication please use the final published version (if applicable).

Samenvatting

Bij complexe problemen uit de praktijk is de eerste stap vaak om ze wiskundig te formuleren, zodat ze toegankelijk worden voor een breed scala aan hulpmiddelen die mogelijk kunnen worden ingezet om ze op te lossen. Onder die wiskundige formuleringen bevindt zich een gevreesde groep: de niet-deterministische polynomiale-tijd-moeilijke (NP-moeilijke) problemen, die doorgaans als onoplosbaar in polynomiale tijd worden beschouwd. NP-moeilijke problemen hebben twee belangrijke kenmerken: efficiënt een juiste oplossing vanaf nul bedenken is doorgaans onmogelijk, maar controleren of een oplossing juist is, is doorgaans makkelijker. Zo ontstonden heuristische methoden: door middel van benaderingen en trial-and-error kunnen ze vaak NP-moeilijke problemen, zelfs grote, binnen een acceptabele tijd oplossen.

Onderzoekers hebben een breed scala aan methoden verzameld om steeds complexere problemen op te lossen. Zodra het juiste hulpmiddel is gekozen – wat op zich al een heel eigen onderzoeksgebied is – is de volgende stap te bepalen hoe dit het best kan worden ingezet. Heuristische algoritmen beschikken over diverse parameters die te vergelijken zijn met schakelaars en knoppen waarmee men kan spelen totdat de snelste weg naar een oplossing is gevonden. Ze maken het bijvoorbeeld mogelijk om voorbewerkingsstappen te activeren, te kiezen uit verschillende meetmethoden, te bepalen in welke richting een zoekproces zich eerst moet vertakken, of de kans te definiëren om vanaf een willekeurige positie opnieuw te starten. Dit alles kan worden afgestemd op de kenmerken van de betreffende probleeminstanties, en het correct instellen ervan kan het verschil maken tussen het oplossen van een probleem in minuten, uren of weken. Het optimaliseren van de prestaties van dergelijke algoritmen door de waarden van hun parameters in te stellen, is een optimalisatieprobleem dat bekend staat als het geautomatiseerde algoritmeconfiguratieprobleem. In de afgelopen twintig jaar hebben onderzoekers heuristische algoritmen (configurators genoemd) ontwikkeld om hun heuristische algoritmen (solvers genoemd) te configureren. Optimalisatiemetho-

den die zijn gebaseerd op verschillende paradigma's — zoals evolutionaire algoritmen, racing en Bayesiaanse optimalisatie — hebben de basis gelegd voor het configureren van de interne werking van algoritmen uit diverse toepassingsgebieden. Die configuratoren, en meer specifiek hun strategieën voor het kiezen van parameterwaarden en probleeminstanties, vormen het hoofdonderwerp van dit proefschrift. We zijn met name geïnteresseerd in de toepassing ervan om de looptijd van NP-moeilijke solvers te verkorten.

We beginnen met het in kaart brengen van de stand van zaken, waarbij we 7 verschillende benaderingen evalueren op 20 datasets en configuratiescenario's uit de literatuur. We constateren dat configuratoren verschillend presteren afhankelijk van de taken, maar ook dat recentere benaderingen niet altijd significant beter presteren dan oudere. Dit toont aan dat er de afgelopen jaren beperkte vooruitgang is geboekt in de stand van zaken op het gebied van looptijdoptimalisatie. Veel recente methoden zijn namelijk afgestemd op het specifieke probleem van hyperparameteroptimalisatie voor machine learning-modellen, waarbij de nadruk ligt op de kwaliteit van de oplossing. We hebben ook de kenmerken van de configuratiescenario's op een rijtje gezet en geen duidelijk verband gevonden tussen deze kenmerken en de prestaties van de verschillende configuratoren. Bepalen welke configurator het beste is voor welk scenario is op zich al een lastig probleem.

Vervolgens richten we onze aandacht op de steekproefverdeling van nieuwe parameterwaarden rond de standaardwaarde. Deze waarde wordt doorgaans opgegeven bij de NP-moeilijke solvers die in ons onderzoek van belang zijn. We constateren dat deze waarden, die doorgaans door de ontwikkelaar worden vastgesteld op basis van hun eigen inzicht in de algoritmen of op basis van experimenten met hun eigen reeks benchmarks, een sterke a-priori-verdeling opleveren bij het genereren van nieuwe waarden. We laten zien dat we met behulp van deze a-priori-informatie de prestaties kunnen verbeteren van de state-of-the-art configuratoren op basis van Bayesiaanse optimalisatie, SMAC.

Na het kiezen van parameterwaarden zou de configurator de probleeminstanties kiezen waarop die parameterwaarden getest moeten worden. We hebben verschillende benaderingen voor instantieselectie ontwikkeld op basis van de literatuur om te bepalen welke instanties de solver als eerste moet proberen op te lossen, en om snel vast te stellen of een set parameterwaarden tot betere prestaties leidt. We evalueren deze in verschillende gevallen: door twee algoritmen te vergelijken, twee geparametriseerde versies van één algoritme, en op basis van verschillende hoeveelheden en soorten a-priori-informatie om verschillende stappen van het configuratieproces in onze experi-

menten weer te geven. Vervolgens hebben we ze geïntegreerd in SMAC en deze nieuwe versie van SMAC met instantieselectie geëvalueerd. We bereikten betere prestaties dan nooit eerder waren bereikt. Het is echter moeilijk te bepalen welke methode wanneer moet worden gebruikt, en deze methoden vereisen daarom meer onderzoek voordat ze in de praktijk toepasbaar zijn.

In deze scriptie hebben we verschillende methoden geëvalueerd om zowel nieuwe parameterwaarden als nieuwe probleeminstanties te genereren op basis van voorafgaande informatie, afkomstig van de ontwikkelaar of uit de tot nu toe verzamelde gegevens. We hebben aangetoond dat configuratietools efficiënter kunnen werken als deze informatie beter wordt benut.

