Lista zadań nr 5

Mikołaj Słupiński

grudzień 2020

Zadanie 1. ***(4 punkty)***

(a) Zapoznaj się z notebookiem zawierającym częściową implementację algorytmu CMA-ES

(b) Uzupełnij luki w implementacji

(c) Sporządź wykresy ilustrujące działanie algorytmu dla funkcji sferycznej, cigar, ellipsoid oraz Rastrigina o wymiarowości 2 dla kilku (np. 5) generacji, tzn. dla generacji g

- Wypunktuj wylosowane osobniki, osobnym kolorem osobniki odrzucone, oraz osobniki wybrane
- Narysuj obrys elipsy odpowiadającej 95% obszarowi prawdopodobieństwa rozkładu normalnego $N(m^{(g)}, C^{(g)})$
- Osobnym kolorem obrys elipsy odpowiadającej 95% obszarowi prawdopodobieństwa rozkładu normalnego $N(m^{(g+1)}, C^{(g+1)})$

Zadanie 2. (3 punkty)

(a) Użyj CMA-ES dla funkcji benchmarkowych zawartych w notebooku o wymiarowości d = 5, 10, 20, 50.

(b) Sprawdź wpływ parametrów algorytmów na jego działanie.

(c) Porównaj działanie CMA-ES z $(\mu + \lambda)$-ES (możesz wykorzystać implementację z poprzedniej listy)

UWAGA: *W trakcie obliczeń na funkcjach o większej wymiarowości mogą zdarzyć się błędy w obliczeniach numerycznych. W rezultacie mogą pojawić się liczby zespolone. Aby tego uniknąć w skrypcie dodano własną implementację obliczania $C^{-1/2}$, proszę z niej korzystać. Również zwiększenie wartości rozmiaru kroku dla dużych wymiarów przynosi pozytywne rezultaty*

Zadanie 3. (3 punkty) Algorytm CMA-ES możemy zmodyfikować tak, aby wagi nie musiały sumować się do jedynki oraz niekoniecznie były dodatnie. Idea tej modyfikacji polega na przypisaniu rozwiązaniam „dobrym" dodatnich wag, natomiast „złym" wag ujemnych.

W oryginalnej wersji algorytmu CMA-ES macierz kowariancji jest aktualizowana wedlug poniższego wzoru:

$$\mathbf{C} = (1 - c_{cov})\mathbf{C} + c_{cov}(\alpha_{cov}\mathbf{p_C}\mathbf{p_C}^{\mathrm{T}} + (1 - \alpha_{cov})\mathbf{Z})$$

gdzie

$$\mathbf{Z} = \mathbf{BD}\left(\frac{1}{\mu}\sum_{k=1}^{\mu}\mathbf{z}_{k;\lambda}\mathbf{z}_{k;\lambda}^{\mathrm{T}}\right)(\mathbf{BD})^{\mathrm{T}}$$

oraz

$$c_{\mathrm{cov}} = \alpha_{cov}\frac{2}{(n+\sqrt{2})^2} + (1 - \alpha_{cov})\min(1, \frac{2\mu - 1}{(n+2)^2 + \mu})$$

gdzie $\alpha_{cov} \in [0, 1]$.

Kiedy chcemy zastosować ujemne wagi powyższe wzory przekształcamy w następujący sposób:

$$\mathbf{Z} = \mathbf{BD}\left(\tfrac{1}{\mu}\textstyle\sum_{k=1}^{\mu}\mathbf{z}_{k;\lambda}\mathbf{z}_{k;\lambda}^{\mathrm{T}} \; -\tfrac{1}{\mu}\sum_{k=\lambda-\mu+1}^{\lambda}\mathbf{z}_{k;\lambda}\mathbf{z}_{k;\lambda}^{\mathrm{T}}\right)(\mathbf{BD})^{\mathrm{T}} \;\;,$$

$$\mathbf{C} = (1 - c_{\mathrm{cov}})\,\mathbf{C} + c_{\mathrm{cov}}\mathbf{p_C}\mathbf{p_C}^{\mathrm{T}} + \beta\mathbf{Z}$$

oraz

$$c_{\mathrm{cov}} = \frac{2}{(n+\sqrt{2})^2},$$
$$\beta = \frac{4\mu - 2}{(n+12)^2 + 4\mu}$$

(a) Zaimplementuj powyższą strategię ewolucyjną

(b) Przetestuj jej działanie na funkcjach wymienionych powyżej. Kiedy radzi sobie istotnie lepiej niż CMA-ES?

Modyfikacja ta jest powszechnie znana i nosi nazwę Active CMA-ES[1].

Zadanie 4. (nieobowiązkowe, 4 punkty)

(a) Zapoznaj się z implementacją autora algorytmu dostępną pod adresem `https://github.com/CMA-ES/pycma`

(b) Porównaj działanie implementacji dostarczonej w notebooku do implementacji autora algorytmu. W tym celu wielokrotnie powtórz obliczenia (algorytm jest randomizowany) i zmierz czas wykonania oraz szybkość zbieżności. Spróbuj wyjaśnić różnice.

[1] *Improving Evolution Strategies through Active Covariance Matrix Adaptation Grahame* A. Jastrebski and Dirk V. Arnold