Wrocław 16.06.2023

Sprawozdanie Projekt PPD

Optymalizacja strategii wyścigu w Formule 1

Jakub Piotrowski; 266502

Cel: Głównym celem projektu jest zastosowanie modelowania i rozwiązywania problemów optymalizacyjnych dla strategii wyścigowej Formuły 1. Problem polega na optymalnym doborze zestawów opon i liczby pit-stopów, aby zminimalizować zużycie paliwa i czas przejazdu.

Opis problemu:

Formuła 1, znana również jako F1, jest najwyższym poziomem międzynarodowego sportu motorowego, uznawanym przez Międzynarodową Federację Samochodową (FIA). Składa się z serii wyścigów, zwanych Grand Prix, które odbywają się na całym świecie na różnego rodzaju torach. Wyścigi te są niezwykle złożonymi przedsięwzięciami, które wymagają nie tylko niezwykłych umiejętności kierowcy, ale także starannego planowania i zarządzania przez zespoły techniczne.

Jednym z najważniejszych aspektów zarządzania wyścigiem jest strategia dotycząca opon i zużycia paliwa. Opony F1 różnią się pod względem miękkości (Soft, Medium, Hard), a wybór odpowiedniego typu opon może mieć znaczny wpływ na wydajność pojazdu. Miękkie opony oferują większą przyczepność, ale szybciej się zużywają, podczas gdy twardsze opony są mniej przyczepne, ale dłużej wytrzymują. Ponadto, ilość dostępnego paliwa wpływa na wagę pojazdu, a co za tym idzie, na czas okrążenia.

Wybór odpowiedniej strategii opon i paliwa to klucz do sukcesu. Musi to uwzględniać wiele czynników, takich jak liczba okrążeń, zużycie opon, zużycie paliwa, czas trwania pit-stopu i wiele innych. Celem jest zminimalizowanie czasu wyścigu poprzez optymalny dobór zestawów opon i ilości pit-stopów, jednocześnie zwracając uwagę na dostępne zasoby, takie jak paliwo.

Mój projekt skupia się na modelowaniu i optymalizacji tej skomplikowanej strategii wyścigu. Stworzony model matematyczny uwzględnia różne parametry i ograniczenia, w tym średni czas okrążenia na różnych typach opon, dostępną ilość paliwa, czas trwania pit stopu, zużycie paliwa na okrążenie dla poszczególnych typów opon i kary za przekroczenie zalecanej ilości okrążeń na danym zestawie opon.

Jest to skomplikowany problem optymalizacyjny, który pozwala zespołom F1 lepiej zrozumieć i zarządzać strategią wyścigu, co ostatecznie przekłada się na ich konkurencyjność na torze.

Parametry:

- sredni_czas_okrazenia_S, sredni_czas_okrazenia_M, sredni_czas_okrazenia_H - średnie czasy okrążenia na oponach typu Soft, Medium i Hard.
- N - liczba okrążeń do pokonania.
- petrol - dostępna ilość paliwa.
- tpit - czas trwania pit stopu.
- zuzycie_paliwa_S, zuzycie_paliwa_M, zuzycie_paliwa_H - tablice zawierające zużycie paliwa na okrążenie dla poszczególnych typów opon.
- kara_S, kara_M, kara_H - tablice zawierające wartości kar za przekroczenie zalecanej ilości okrążeń na danym zestawie opon.
- scieranie_S, scieranie_M, scieranie_H - ilość okrążeń, po której opony danego typu ulegają zbyt dużemu zużyciu.

Zmienne decyzyjne:

- xS, xM, xH - liczba okrążeń na oponach danego typu.
- pitstop_S, pitstop_M, pitstop_H - liczba pitstopów na oponach danego typu.

Funkcja celu:

Celem jest minimalizacja czasu przejazdu, uwzględniając średni czas okrążenia dla każdego typu opon, kary za przekroczenie limitu okrążeń dla danego zestawu opon i czas trwania pit stopu.

Ograniczenia:

- Łączna liczba okrążeń musi być równa sumie okrążeń na poszczególnych oponach.
- Suma zużycia paliwa na poszczególnych oponach musi mieścić się w dostępnym paliwie, z zachowaniem zasady pozostawienia co najmniej 1 litra na koniec wyścigu.
- Liczba pitstopów na każdym typie opon musi być większa lub równa wynikowi dzielenia liczby okrążeń przez średnią trwałość opon danego typu, pomniejszonej o 1.

Ten model optymalizacyjny pomaga w zrozumieniu, jak optymalnie zarządzać zużyciem paliwa i wyborem opon w trakcie wyścigu Formuły 1, mając na uwadze zmienne takie jak zużycie opon, średni czas okrążenia, zużycie paliwa i liczba pitstopów.

Kod CPLEX:

```
float sredni_czas_okrazenia_S = 10;
float sredni_czas_okrazenia_M = 10.5;
float sredni_czas_okrazenia_H = 11.0;
int N = 50;
float petrol = 90.0;
float tpit = 2.5;
float zuzycie_paliwa_S[1..N] = [0.9, 0.89, 0.88, 0.87, 0.86, 0.85, 0.84, 0.83, 0.82,
0.81, 0.8, 0.79, 0.78, 0.77, 0.76, 0.75, 0.74, 0.73, 0.72, 0.71, 0.7, 0.69, 0.68, 0.67,
0.66, 0.65, 0.64, 0.63, 0.62, 0.61, 0.6, 0.59, 0.58, 0.57, 0.56, 0.55, 0.54, 0.53, 0.52,
0.51, 0.5, 0.49, 0.48, 0.47, 0.46, 0.45, 0.44, 0.43, 0.42, 0.41];
float zuzycie_paliwa_M[1..N] = [0.8, 0.79, 0.78, 0.77, 0.76, 0.75, 0.74, 0.73, 0.72,
0.71, 0.7, 0.69, 0.68, 0.67, 0.66, 0.65, 0.64, 0.63, 0.62, 0.61, 0.6, 0.59, 0.58, 0.57,
0.56, 0.55, 0.54, 0.53, 0.52, 0.51, 0.5, 0.49, 0.48, 0.47, 0.46, 0.45, 0.44, 0.43, 0.42,
0.41, 0.4, 0.39, 0.38, 0.37, 0.36, 0.35, 0.34, 0.33, 0.32, 0.31];
float zuzycie_paliwa_H[1..N] = [0.7, 0.69, 0.68, 0.67, 0.66, 0.65, 0.64, 0.63, 0.62,
0.61, 0.6, 0.59, 0.58, 0.57, 0.56, 0.55, 0.54, 0.53, 0.52, 0.51, 0.5, 0.49, 0.48, 0.47,
```

```
0.46, 0.45, 0.44, 0.43, 0.42, 0.41, 0.4, 0.39, 0.38, 0.37, 0.36, 0.35, 0.34, 0.33, 0.32,
0.31, 0.3, 0.29, 0.28, 0.27, 0.26, 0.25, 0.24, 0.23, 0.22, 0.21];
float kara_S[1..N] =
[0,0,0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,13,14,14,15,15,16,
16,17,17,18,18,19,19,20,20,21,21,22,22,23];
float kara_M[1..N] =
[0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,6,6,6,6,7,7,7,7,8,8,8,8,9,9,9,9,10,
10,10,10,11,11,11];
float kara_H[1..N] =
[0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6
,7,7,7,7,7];
int scieranie_S = 5;
int scieranie_M = 7;
int scieranie_H = 9;
dvar int+ xS;
dvar int+ xM;
dvar int+ xH;
dvar int+ pitstop_S;
dvar int+ pitstop_M;
dvar int+ pitstop_H;
minimize
   xS * sredni_czas_okrazenia_S + sum(i in 1..N) maxl(0, xS - i)*kara_S[i] - sum(i in
1..N) maxl(0, xS-i-1)*kara_S[i]
   + xM * sredni_czas_okrazenia_M + sum(i in 1..N) maxl(0, xM - i)*kara_M[i] - sum(i
in 1..N) maxl(0, xM-i-1)*kara_M[i]
   + xH * sredni_czas_okrazenia_H + sum(i in 1..N) maxl(0, xH - i)*kara_H[i] - sum(i
in 1..N) maxl(0, xH-i-1)*kara_H[i]
   + (pitstop_S + pitstop_M + pitstop_H)*tpit;
subject to {
   xS + xM + xH == N;
   sum(i in 1..N) maxl(0, xS - i)*zuzycie_paliwa_S[i] - sum(i in 1..N) maxl(0, xS - i -
1)*zuzycie_paliwa_S[i]
   + sum(i in 1..N) maxl(0, xM - i)*zuzycie_paliwa_M[i] - sum(i in 1..N) maxl(0, xM - i -
1)*zuzycie_paliwa_M[i]
```

```
    + sum(i in 1..N) maxl(0, xH - i)*zuzycie_paliwa_H[i] - sum(i in 1..N) maxl(0, xH - i -
1)*zuzycie_paliwa_H[i] <= petrol-1;
    pitstop_S >= (xS / scieranie_S)-1;
    pitstop_M >= (xM / scieranie_M)-1;
    pitstop_H >= (xH / scieranie_H)-1;
}
```

Metaheurystyka - Optymalizacja Rojem Cząstek (PSO)

W ramach tego projektu zastosowano metaheurystykę Optymalizacji Rojem Cząstek (PSO) w celu rozwiązania problemu optymalnego doboru strategii wyścigowej Formuły 1. PSO jest algorytmem inspirowanym zachowaniem stad zwierząt, w którym populacja cząstek (rozwiązania) porusza się po przestrzeni poszukiwań w celu znalezienia optymalnego rozwiązania.

Algorytm PSO został zaimplementowany przy użyciu biblioteki pyswarms w języku Python. Biblioteka pyswarms dostarcza narzędzia do łatwej implementacji różnych wariantów PSO oraz zapewnia wygodny interfejs do definiowania funkcji celu, ograniczeń i innych parametrów.

Wyniki

Przeprowadzono symulacje w CPLEX i pythonie używając tych samych parametrów

CPLEX optimization studio:

```
// solution (optimal) with objective 585.5
// Quality Incumbent solution:
// MILP objective                                  5,8550000000e+02
// MILP solution norm |x| (Total, Max)             1,05210e+04  4,10000e+01
// MILP solution error (Ax=b) (Total, Max)         6,66134e-15  6,66134e-15
// MILP x bound error (Total, Max)                 0,00000e+00  0,00000e+00
// MILP x integrality error (Total, Max)           0,00000e+00  0,00000e+00
// MILP slack bound error (Total, Max)             0,00000e+00  0,00000e+00
//

xS = 10;
xM = 18;
xH = 22;
pitstop_S = 1;
pitstop_M = 2;
pitstop_H = 2;
```

Metaheurystyka

Python:

```
2023-06-19 23:08:16,228 - pyswarms.single.global_best - INFO - Optimize 
pyswarms.single.global_best: 100%|██████████|16000/16000, best_cost=620
Najlepsze rozwiązanie:  [ 8 18 24  4  6  4] , koszt:  620.0
2023-06-19 23:08:21,755 - pyswarms.single.global_best - INFO - Optimizat
```

xS = 8

xM = 18

xH = 24

pitstop_S = 4

pitstop_M = 6

pitstop_H = 4

f(x) = 620

Wnioski

Niniejszy projekt ilustruje praktyczne zastosowanie technik modelowania i optymalizacji w rozwiązywaniu złożonych problemów rzeczywistych, takich jak strategia wyścigu w Formule 1. Istotnym elementem analizy było uwzględnienie wielu zmiennych wpływających na wynik wyścigu, w tym typu opon, liczby pit-stopów, zużycia paliwa oraz liczby okrążeń.

Porównując strategie wypracowane za pomocą Pythona i narzędzia optymalizacyjnego CPLEX, wyraźnie zauważyć można wyższą efektywność tego drugiego. Python sugerował strategię składającą się z większej liczby pit-stopów, szczególnie na oponach typu Soft i Medium. Z kolei, CPLEX proponował strategię z mniejszą liczbą pit-stopów, co skutkowało krótszym czasem trwania wyścigu. Różnica w wartościach funkcji celu (630 dla Pythona w porównaniu do 585,5 dla CPLEX) dowodzi wyższej efektywności strategii zaproponowanej przez CPLEX. Dodatkowym atutem CPLEX jest też krótszy czas obliczeń (3,22s), w porównaniu do Pythona (5,527s).

Te rezultaty mają kluczowe znaczenie dla strategii wyścigów Formuły 1, umożliwiając zespołom lepsze zrozumienie i zarządzanie strategią wyścigową, co bezpośrednio przekłada się na ich konkurencyjność na torze.

Podsumowując, wybór narzędzi do rozwiązania problemu optymalizacyjnego powinien uwzględniać specyfikę problemu i dostępne zasoby. Zaawansowane techniki modelowania i optymalizacji, jak pokazał ten projekt, mogą znacząco usprawnić proces podejmowania decyzji, nawet w tak skomplikowanych dziedzinach, jak strategia wyścigowa w Formule 1.