

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

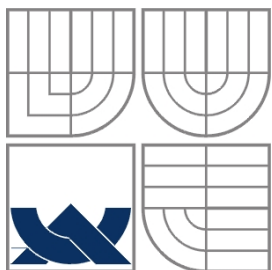
OPTIMALIZACE NASTAVENÍ ZÁVODNÍHO VOZU
SIMULÁTORU TORCS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

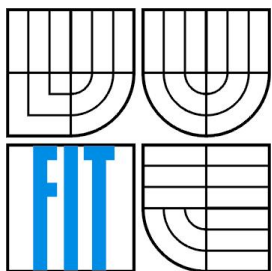
AUTOR PRÁCE
AUTHOR

Bc. Pavel Srnec

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

OPTIMALIZACE NASTAVENÍ ZÁVODNÍHO VOZU SIMULÁTORU TORCS

OPTIMIZATION OF A RACING CAR SETUP WITHIN TORCS SIMULATOR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavel Srnec

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Petr Pospíchal

BRNO 2012

Abstrakt

Tato práce pojednává o přírodních optimalizačních technikách, které jsou využity pro řešení problému nastavení závodního vozu. Nejdříve jsou v rámci optimalizačních technik představeny evoluční algoritmy. Největší důraz je kladen na optimalizaci hejnem částic. Nastavení závodního vozu je popsáno v samostatné kapitole spolu s představením simulátoru TORCS. Práce obsahuje veškeré návrhové a implementační části zvolených technik. Hlavním cílem práce je nalezení optimálního nastavení pro různé specifikace okruhů.

Abstract

This master's thesis is about nature optimization techniques. Evolution algorithms together with main thesis topic, Particle Swarm Optimization, is introduced in the following chapter. Car setup and simulator TORCS are introduced in next chapter. Design and implementation are introduced in next chapters. Destination of the master's thesis is finding optimal car setups for different circuits.

Klíčová slova

Optimalizace, PSO, TORCS, závodní vůz, optimalizační techniky, nastavení vozu, genetické algoritmy, optimalizace hejnem částic

Keywords

Optimization, PSO, TORCS, racing car, optimization techniques, car setup, genetic algorithms, particle swarm optimization

Citace

Pavel Srnec: Optimalizace nastavení závodního vozu simulátoru TORCS, diplomová práce, Brno, FIT VUT v Brně, 2012

Optimalizace nastavení závodního vozu simulátoru TORCS

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Petr Pospíchala. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Srnec
18.05.2012

Poděkování

Chtěl bych poděkovat Ing. Pospíchalovi za odborné vedení této práce a také Bc. Pavle Kůrkové za poskytnutí materiálů k problematice Inteligence hejna.

© Pavel Srnec, 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod.....	4
2 Optimalizační techniky.....	5
2.1 Evoluční algoritmy.....	5
2.1.1 Darwinova teorie.....	5
2.1.2 Evoluční algoritmus.....	5
2.2 Genetické algoritmy.....	6
2.2.1 Zakódování problému a inicializace populace.....	7
2.2.2 Ohodnocení jedinců.....	7
2.2.3 Selektce jedinců.....	8
2.2.4 Křížení.....	8
2.2.5 Mutace.....	9
2.2.6 Obnova populace.....	9
2.3 Optimalizace hejnem částic.....	10
2.3.1 Inteligence hejna.....	10
2.3.2 Algoritmus optimalizace hejnem částic.....	11
2.3.3 Inicializace jedinců v prohledávaném prostoru.....	11
2.3.4 Interakce a pohyb jedinců.....	12
2.3.5 Ohodnocení jedinců.....	13
2.4 Hybridní algoritmus.....	13
2.5 Shrnutí.....	14
3 Nastavení vozu.....	15
3.1 Obecný přehled.....	15
3.2 Závodní informace.....	15
3.2.1 Závodní série.....	16
3.2.2 Povrch trati.....	16
3.2.3 Formát závodního víkendu.....	16
3.2.4 Prvotní nastavení.....	17
3.3 Parametry závodního vozu.....	17
3.3.1 Brzdy.....	18
3.3.2 Pružiny.....	18
3.3.3 Tlumiče.....	19
3.3.4 Příčné stabilizátory.....	19
3.3.5 Světlá výška.....	19

3.3.6 Podvozek.....	20
3.3.7 Přítlačná křídla a difuzor.....	20
3.3.8 Převodovka.....	20
3.3.9 Rozvodovka a diferenciál.....	21
3.3.10 Maximální rejď.....	21
3.3.11 Sbíhavost kol.....	21
3.3.12 Odklon kol.....	22
3.3.13 Rozchod kol.....	22
3.3.14 Geometrie kol.....	23
3.3.15 Pneumatiky.....	23
3.4 Simulátor TORCS.....	23
3.4.1 Popis programu.....	23
3.4.2 Rozšíření TORCS.....	24
3.4.3 Nastavitelné parametry vozu v TORCS.....	25
3.4.4 Popis komunikace.....	25
3.5 Shrnutí.....	26
4 Návrh řešení.....	27
4.1 Představení projektu CIG.....	27
4.2 Ohodnocení výsledků.....	27
4.3 Evoluční algoritmus.....	29
4.3.1 Zakódování problému nastavení vozu.....	29
4.3.2 Počet jedinců a iterací algoritmu.....	29
4.3.3 Selektce, křížení, mutace a obnova populace.....	29
4.4 Optimalizace hejnem částic.....	30
4.4.1 Prohledávaný prostor, počet jedinců a komunikace.....	30
4.4.2 Volba koeficientů.....	31
4.5 Hybridní algoritmus.....	31
4.6 Ukládání výsledků.....	31
4.7 Shrnutí.....	32
5 Implementace.....	33
5.1 Komunikace.....	33
5.2 Fitness funkce.....	34
5.3 Optimalizační algoritmus.....	34
5.4 Genetický algoritmus.....	35
5.5 Optimalizace hejnem částic.....	36

5.6 Hybridní algoritmus.....	37
5.7 Shrnutí.....	38
6 Dosažené výsledky.....	39
6.1 Obecné zhodnocení.....	39
6.2 Asfaltové okruhy.....	41
6.2.1 Forca.....	41
6.2.2 CG Speedway number 1.....	42
6.2.3 CG track 2.....	43
6.2.4 Olethros Road 1.....	44
6.2.5 Street 1.....	45
6.2.6 Wheel 2.....	46
6.2.7 Alpine 2.....	47
6.3 Ovály.....	47
6.3.1 C-Speedway.....	48
6.3.2 Michigan Speedway.....	49
6.4 Štěrkové okruhy	49
6.4.1 Dirt 1.....	50
6.4.2 Dirt 3.....	50
6.4.3 Dirt 5.....	51
6.4.4 Mixed 1.....	52
6.5 Shrnutí.....	53
7 Závěr.....	54
Literatura.....	55
Příloha 1 - Zprovoznění projektu.....	57
Windows.....	57
Linux.....	57
Příloha 2 – Nalezená nastavení.....	59
Seznam příloh.....	66

1 Úvod

Práce pojednává o optimalizačních technikách inspirovaných přírodou, které jsou využity k řešení problému nastavení závodního vozu. K vyhodnocení slouží simulátor The Open Racing Car Simulator (dále TORCS).

Podstata práce vychází z projektu Milánské univerzity CIG (Computational Intelligence in Games) [21] a konference GECCO (Genetic and Evolutionary computation conference) [22]. Práce se zabývá variantou projektu vyhodnocení nastavení závodního vozu na jedno kolo libovolného okruhu simulátoru TORCS. Druhou variantou je naučení řízení vozu pro libovolný závodní úsek.

Na začátku práce jsou představeny základní optimalizační techniky v podobě evolučních algoritmů. Evoluční algoritmy jsou zvoleny především v rámci uvedení do problematiky optimalizace přírodou inspirovaných technik. Také proto obsahují velkou část kapitoly 2 Optimalizační techniky. Poté je představena technika optimalizace hejnem částic (Particle Swarm Optimization). Tato optimalizace je zvolena pro své vlastnosti vzhledem k problému nastavení závodního vozu, které může mít více správných výsledků (kombinace parametrů vozu).

Nastavení závodního vozu je další důležitá část práce. Problematika je uvedena v rámci kapitoly 3 Nastavení vozu. Tato kapitola obsahuje obecné informace a principy nastavení závodního vozu. Nezachází zbytečně do detailů, jelikož problematika nastavení není uplatněna v implementačním řešení diplomové práce. Ovšem znalost nastavení jednotlivých parametrů je důležitá v rámci návrhu a vyhodnocení výsledků, které vrátí optimalizace. Jinak řečeno znalost problému nastavení je důležitá z důvodu správného zadání vstupu a objasnění výstupu.

Kapitola následující již obsahuje část práce, která se věnuje návrhu řešení problému nastavení vozu. Návrh řešení je uveden pro každou optimalizační techniku uvedenou v kapitole 2. Pozornost je zaměřena na genetické algoritmy a optimalizaci pomocí částicových hejn.

Následuje představení algoritmů po implementační stránce. V krátkosti je uvedena komunikace a poté rozebrání algoritmů, kde jsou uvedeny podstatné části zdrojových kódů.

Nejdůležitější kapitolou je v pořadí šestá Dosažené výsledky. Zde jsou představeny okruhy, které se testovaly v rámci hledání optimálního nastavení. Ve stručnosti je uvedeno pár postřehů, které by bylo vhodné zohlednit při klasickém postupu nastavení závodního vozu. Ihned poté jsou shrnuty výsledky optimalizačních algoritmů.

2 Optimalizační techniky

Optimalizační techniky použité v této práci jsou inspirované výhradně přírodou. Hlavní důraz je kladen na evoluční algoritmy a optimalizaci hejnem částic. Pro doplnění byl vybrán hybridní algoritmus zakázaného prohledávání. Pro lepší názornost a uvedení do problematiky se začíná vysvětlením evolučních algoritmů.

2.1 Evoluční algoritmy

Evoluční algoritmy (kapitola čerpala především z [5] a [2]) jsou stochastická metoda optimalizace pracující s populací jedinců. Je zadán určitý problém a v rámci prohledávání množiny možných řešení se hledá optimální řešení. Jednotlivá kandidátní řešení jsou postupně vylepšována. Evoluční algoritmy vycházejí z Darwinovy teorie.

2.1.1 Darwinova teorie

Darwinova teorie [5] založená na přirozeném výběru druhů, vysvětluje adaptaci populace na prostředí, ve kterém se vyskytuje. Populace je schopna zvětšovat svůj počet do nekonečna, ale je omezená okolím, respektive zdroji nacházející se v daném prostředí. Jedinci, kteří jsou schopni obstarat si snadněji potravu, se stávají silnějšími. Zatímco slabší jedinci umírají, silnější jsou přirozeně vybráni k rozmnožení. Takto se daná populace jedinců adaptuje na prostředí, ve kterém žije a stává se odolná (postupně si zvyká) na případné změny prostředí. Noví jedinci dostávají genetické informace od rodičů. Projevuje se důležitá vlastnost dědičnosti.

Darwinovu teorii doplnil Mendel tvrzením, že evoluce probíhá přirozeným výběrem selekce a náhodných genetických mutací.

Důležitým pojmem pro evoluční algoritmy je tzv. hodnota fitness. Jedná se o ohodnocení každého jedince na základě schopnosti přežít v daném okolí.

2.1.2 Evoluční algoritmus

Evoluční algoritmy [2] jsou založeny na myšlence evoluce, kdy je hledáno globální maximum (minimum) na základě nejvyšší (nejnižší) hodnoty fitness jedince reprezentujícího řešení problému.

Ovšem na rozdíl od evoluce je nutné algoritmus určitým způsobem ukončit. Záleží na znalosti parametrů výsledného řešení. Pokud není možné z této znalosti sestavit podmínku signalizující konečné řešení, je možné algoritmus ukončit jednou z daných podmínek:

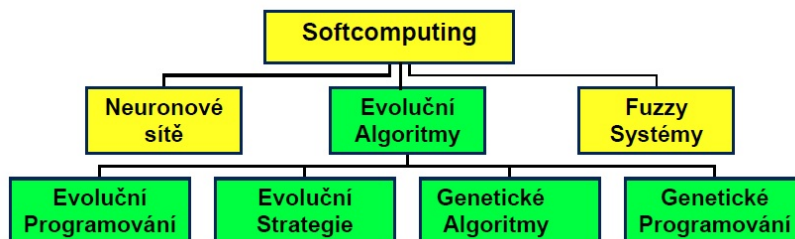
- 1) Předem daným počtem generací populace (v čase t).

- 2) Při neměnné hodnotě fitness po určitý počet generací
- 3) Dosažením uspokojivého řešení.

Problémem je nejistota dosažení globálního extrému funkce. Pokud je možnost sestavit výše zmíněnou podmínku, algoritmus se ukončí po jejím splnění (případně podobně jako v předchozím případě).

Evoluční algoritmus se řídí následujícími kroky:

- 1) Vhodné zakódování problému do řetězce symbolů.
- 2) Náhodné vygenerování počáteční populace.
- 3) Vygenerované jedince je zapotřebí ohodnotit jejich fitness hodnotou.
- 4) Pokud je splněna podmínka ukončení, algoritmus se ukončí.
- 5) Inkrementuje se číslo populace.
- 6) Proběhne výběr jedinců (rodičů) pro operátory křížení a mutace.
- 7) Rekombinace vybraných jedinců.
- 8) Aplikuje se náhodná mutace potomků.
- 9) Ohodnocení nově vzniklých jedinců.
- 10) Obnova populace a pokračování bodem 4)



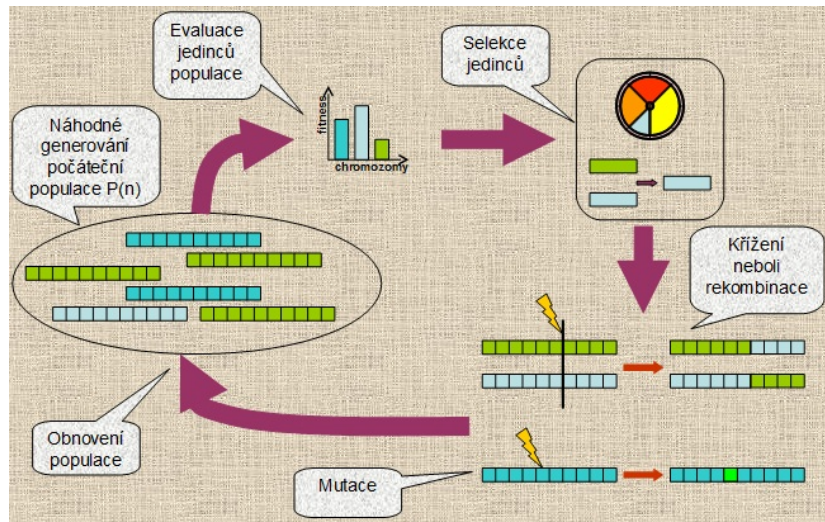
Obrázek 2.1 Softcomputing - dělení [5]

Evoluční algoritmy se dělí na různé varianty. Samy jsou podoblastí Softcomputingu¹ (obrázek 2.1). Každá z těchto variant řeší jednotlivé kroky odlišným způsobem. Pozornost bude zaměřena na nejrozšířenější oblast Evolučních algoritmů a sice na Genetické algoritmy.

2.2 Genetické algoritmy

Uplatnění genetických algoritmů [5] je při hledání globálního extrému funkce, kdy je obtížné (případně nemožné) nalézt analytické řešení. Genetické algoritmy se vyznačují použitím operátoru křížení (v předchozí kapitole bod 7 Rekombinace vybraných jedinců) jako hlavním tahounem v hledání globálního extrému. Běh genetického algoritmu je zobrazen na obrázku 2.2. V dalších podkapitolách jsou uvedeny jeho jednotlivé části.

¹ Oproti klasickým metodám akceptuje nepřesnosti reálného světa.



Obrázek 2.2 Genetický algoritmus [30]

2.2.1 Zakódování problému a inicializace populace

Počet jedinců [2] v populaci je závislý na náročnosti problému. Obvykle se pohybuje okolo 100 jedinců. V případě náročnějšího problému, kdy se předpokládá větší množina možných řešení, je možné počet zvýšit na 200 jedinců.

Prvotní populace je vygenerována náhodně, z důvodu pokrytí co největšího spektra možných řešení. V určitých případech je možné algoritmu pomoci. Například v případě permutačních problémů je možné, již v inicializaci uvažovat permutační zakódování. Zakódování problému je velice důležité pro správnou funkčnost algoritmu.

2.2.2 Ohodnocení jedinců

Ohodnocení jedinců [5] probíhá pomocí fitness funkce. Každému jedinci je přidělena jeho vlastní fitness hodnota. V případě hledání globálního maxima je nejlepší jedinec s maximální možnou fitness. Hledání globálního minima je nejlepší jedinec ohodnocen minimální fitness hodnotou.

Bohužel neexistuje přesně daný předpis pro vytvoření fitness funkce. Obecný postup výpočtu probíhá z obsahu aktuálního potomka v generaci, kdy je snaha z těchto informací vypočítat fitness hodnotu pomocí znalostí daného problému. Zjednodušeně řečeno, každý problém má svou fitness funkci.

Správně napsaná fitness funkce výrazně ovlivňuje správné nasměrování algoritmu ke globálnímu extrému.

2.2.3 Selekcce jedinců

Selekcce [5] vybírá předem daný počet jedinců v populaci pro potřeby operátorů křížení a mutace. Selekcce si klade za podmínky, vybrání lépe ohodnocených jedinců s větší pravděpodobností. Ovšem nejedná se o pouhé vybrání jedinců pomocí nejlepších fitness hodnot. Je nutné zařídit, aby měli šanci na výběr i hůře ohodnocení jedinci. Tento fakt zajistí pomalejší konvergence algoritmu. Poté je menší pravděpodobnost uvážnutí algoritmu v lokálním extrému, jelikož je zajištěna větší rozmanitost populace.

Jedna z možností selekcce je výběr pomocí rulety. Každému jedinci je přiřazena pravděpodobnostní hodnota pomocí vzorečku 2.1 na základě jeho fitness [5].

$$p_i = \frac{fitness_i}{\sum_{j=0}^N fitness_j} \quad (2.1)$$

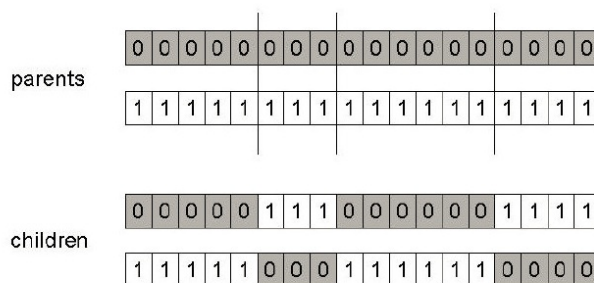
Vzniknou intervaly, jejichž velikost je závislá na hodnotách fitness jednotlivých jedinců. Následně se vygeneruje náhodné číslo od 0 do 1. Jedinec je vybrán na základě náležitosti tohoto čísla do svého intervalu. Tento proces se opakuje, dokud není vybrán požadovaný počet jedinců.

Další možností je turnajová selekcce. Spočívá ve výběru dvojice jedinců, přičemž je vybrán jedinec s větší fitness. Tento proces se opakuje, dokud není vybrán požadovaný počet jedinců. Turnajová selekcce má velice dobré vlastnosti pro správnou konvergenci algoritmu.

2.2.4 Křížení

Operátor křížení [5] je základem genetických algoritmů. Jedná se o výměnu genetických informací od rodičů na nově vzniklé potomky. Operátor křížení se neaplikuje na celý výběr, který vrátí selekcce. Je aplikováno s určitou pravděpodobností, která se volí cca 0,7. Existují různé druhy křížení.

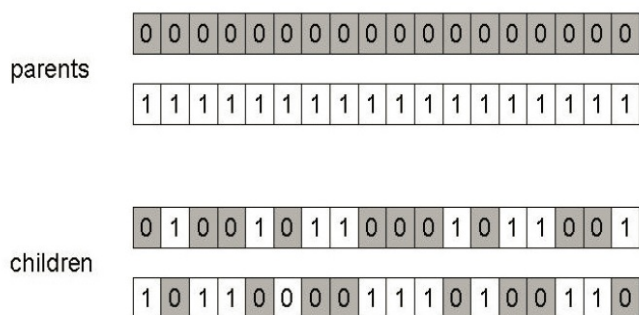
Vícebodové křížení je zobrazeno na obrázku 2.3 (třibodové křížení). Náhodně se vyberou dva jedinci (rodiče) a zvolí se pozice bodů křížení. Minimálně je nutné zvolit jeden bod křížení v rámci jednobodového křížení.



Obrázek 2.3 Vícebodové křížení [5]

Z obrázku 2.3 je patrný výsledek křížení v podobě dvou nových jedinců. Každá dostane kombinaci od obou rodičů. Kombinace jsou odděleny body křížení (na obrázku bílou a šedou barvou).

Další typem je uniformní křížení. Podobně jako u vícebodového křížení jsou výsledkem dva noví jedinci. Pro každý gen se vygeneruje náhodné číslo od 0 do 1. Díky kterému se rozhodne, od kterého rodiče se zkopíruje znak (gen) na totožnou pozici prvního řetězce (genomu). Hodnota „opačného“ rodiče se zkopíruje do druhého řetězce (genomu). Tento proces se opakuje po celou délku řetězce (genomu). Uniformní křížení je naznačeno na obrázku 2.4.



Obrázek 2.4 Uniformní křížení [5]

Uniformní křížení má nevýhodu, jelikož rozvrací stavební bloky genomu. Ovšem výhodou je možnost různorodosti populace.

2.2.5 Mutace

Nejčastější druh mutace [2] je inverze náhodného bitu v genomu. Toto platí pro binární zakódování problému. Obecně pro genetické algoritmy platí, že mutace je náhrada na náhodně vygenerované pozici v genomu náhodnou hodnotou. Nahrazená hodnota je vždy uvažována v rámci řešeného problému. Například pokud se v zakódování vyskytují číselné hodnoty od 1 do 9, je i v operátoru mutace vygenerováno číslo z tohoto intervalu.

Také operátor mutace je aplikován s určitou pravděpodobností, která se pohybuje v rozmezí 0,01-0,1. Mutace je důležitý operátor, který do populace přináší nové informace. Pravděpodobnost nesmí být moc velká, protože by způsobila nestabilitu ve vývoji populace. V případě velmi nízké pravděpodobnosti by se neprojevil vliv nově získaných informací.

2.2.6 Obnova populace

Prvním druhem je úplná obnova [5]. Nově vzniklá populace nahradí populaci původní, která celá zanikne. Druhým typem je částečná obnova populace, kdy se nahradí 20-50% nejslabších jedinců

původní populace, nově vzniklými jedinci, kteří mají vysokou fitness hodnotu. Výběr jedinců je možné provést turnajem (stejný princip jako u selekce).

2.3 Optimalizace hejnem částic

Kapitola obsahuje představení techniky Optimalizace hejnem částic (anglicky Particle Swarm Optimization – PSO). Tato technika vychází z inteligence hejna, která je popsána v následující podkapitole. Dále jsou popsány důležité pojmy PSO. Veškeré informace jsou čerpány z [1], [3], [4], [6], [7], [8] a [9].

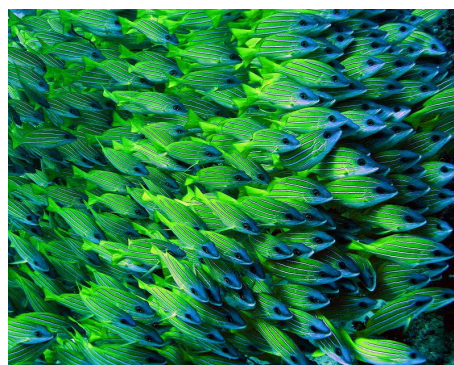
2.3.1 Inteligence hejna

Inteligence hejna [1] je technika založená na chování skupiny, kterou tvoří jedinci s totožnými vlastnostmi. V případě hejna se jedinci rozumí ptáci (obrázek 2.5) nebo ryby (obrázek 2.6).

Jedinci v hejnu se často označují jako agenti. V práci je použit pojem jedinec z důvodu shodného pojmu s kapitolou Evoluční algoritmy.



Obrázek 2.5 Hejno ptáků [31]



Obrázek 2.6 Hejno ryb [32]

Každý jedinec se řídí pravidly, která jsou jednoduchá a není jich mnoho. Například pro hejno ptáků platí „uhni dříve, než dojde ke kolizi,“ „leť stejně rychle jako tvoji sousedé“ a „snaž se letět do pozorovaného cíle.“ Každý nově připojený jedinec udržuje určitou vzdálenost od ostatních jedinců ve svém okolí.

Jedinci se informují o blížícím se nebezpečí lokální komunikací. Například s blížící se překážkou jedinci v přední linii provedou úhybný manévr a v závislosti na svém okolí začnou ostatní jedinci uhýbat také. Ovšem není tomu tak, že by se celé hejno řídilo jedincem (jedinci) vpředu hejna, ale reagují na jedince ve svém okolí. Hejno nemá žádný centralizovaný systém řízení, ale shodné chování všech jedinců v hejnu, vede ke složitému chování celé skupiny, která je schopná řešit rozsáhlé problémy. Popsané chování hejna má určitou podobnost s řízením celulárních automatů (pro další informace [23]).

Chování popsané v předchozím odstavci ukazuje na další důležitou vlastnost hejna a sice samoorganizace. Hejno je odolné vůči odpadnutí jedince (ztracený jedinec, úmrtí jedince), protože pokud jedinec opustí hejno, na jeho místo se posune nejbližší možný. Tento jev je velice důležitý v potřebách robotiky, která se také inspiruje přírodou. Výpadek jednoho robota nemá žádný vliv na celkový systém, protože jej zastoupí jiní nebo se přidá další robot (všichni jsou stejní se stejnými vlastnostmi chování).

Intelligence hejna je technika schopná vyřešit složité problémy za pomoci jednoduchých pravidel. Ovšem důležité je, že řešení nalézt může, ale nemusí. Což platí u všech technik, které souvisí s prohledáváním prostoru (množiny) možných řešení. Jelikož prostor řešení není často prohledán celý (v mnoha případech to není možné).

2.3.2 Algoritmus optimalizace hejnem částic

V podkapitole je představen algoritmus PSO [8], který je popsán v jednotlivých bodech. Podobně jako chování jednotlivce je samotný algoritmus velice jednoduchý.

- 1) Náhodně se vygenerují jedinci v rámci d-dimenzionálního prostoru.
- 2) Podle počtu okolí (počet jedinců pro lokální komunikaci) se náhodně zvolí, kdo s kým bude komunikovat (koho bude mít ve svém okolí).
- 3) Ohodnocení všech jedinců se nastaví na hodnotu 0.
- 4) Nově vzniklé jedinci jsou ohodnoceni.
- 5) V případě lepších výsledků se aktualizují hodnoty $P_{i\text{best}}$ (dosud nejlepší hodnota jedince) a P_G (dosud nejlepší hodnota daného okolí).
- 6) Vypočítají se nové rychlosti a na jejich základě i nové pozice jedinců.
- 7) Pokud není splněna podmínka ukončení, pokračuje se bodem 4)

Podmínkou ukončení je rozuměn určitý počet iterací, případně nalezení uspokojivého řešení daného problému.

2.3.3 Inicializace jedinců v prohledávaném prostoru

V rámci řešení reálného problému je nutné jedince určitým způsobem omezit. Jedinci se pohybují v d-dimenzionálním prostoru kde d náleží N. Počet jedinců n je závislý na počtu dimenzí daného problému. Obvykle se volí podle vzorce 2.2 [6].

$$\text{počet jedinců} = 10 + 2 \sqrt{\text{počet dimenzí}} \quad (2.2)$$

Jedinec je reprezentován pozicí, která představuje vektor $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ kde i je číslo jedince. Vektor X_i je potenciačním řešením problému. Také je potřeba určit velikost okolí, se kterým bude každý jedinec komunikovat. Vzhledem k rozsáhlosti prohledávaného prostoru se okolím rozumí počet jedinců, který je pevně daný. Obvykle se volí od 1 do 5. V případě velkého

počtu dimenzí (větší než 50) je možné volit více jedinců ke komunikaci. Velice často jsou na začátku algoritmu pro každého jedince náhodně vybráni ti, se kterými bude komunikovat v průběhu algoritmu [8].

2.3.4 Interakce a pohyb jedinců

Pro potřeby interakce je zapotřebí ukládání hodnot nejlepších (nalezených) výsledků jedince a okolí, se kterým jedinec komunikuje. Každý jedinec „i“ (pořadí jedince) si pamatuje svoje doposud nejlepší řešení (danou pozici v prostoru) $P_{i\text{best}}$ a nejlepší ohodnocení v rámci okolí, se kterým komunikuje P_G . Pokud určitý jedinec nalezne lepší řešení než je dosavadní P_G , ostatní jedinci v okolí si zapíší hodnotu také [8].

Jedinec „i“ je určený vektorem X_i . Aby bylo vůbec možné prostor prohledávat, je nutný pohyb jedinců. Pohyb není přesně daný (první jedinec se posune o jednotku doleva, druhý doprava), ale není ani náhodný. Pro každého jedince v kroku k je posun do kroku $k+1$ [3]:

$$X_i(k+1) = X_i(k) + V_i(k+1) \quad (2.3)$$

Hodnota V_i (vzorec 2.3) je označována jako rychlost jedince i . Pojmy pozice a rychlost jedince jsou základem metody PSO.

Rychlost jedince reprezentuje také vektor, kde počet dimenzí je shodný s vektorem určujícím pozici. Matematicky lze pohyb jedinců popsat jako součet vektorů. Ovšem je nutné znát rychlost V_i , její výpočet je ve vzorci 2.4, který naznačuje výpočet každé dimenze d vektoru V_i [3]:

$$V_{id}(k+1) = wV_{id}(k) + c_1r_1(P_{id\text{best}} - X_{id}(k)) + c_2r_2(P_{Gd} - X_{id}(k)) \quad (2.4)$$

Ve vzorci 2.4 jsou uvedeny dosud neznámé hodnoty. Jedná se o hodnoty [4] w , c_1 , c_2 , r_1 a r_2 . Všechny jsou koeficienty, které se zvolí na začátku algoritmu a dále zůstávají neměnné². Koeficient w je tzv. setrvačný. Jeho hodnota je reálné číslo z intervalu $<0,1>$. Závisí na ní velikost posunu v následujícím kroku. Pokud bude malá, nemusí se žádný jedinec dostat do globálního optima po celou dobu běhu algoritmu. Naopak pokud se bude blížit k hodnotě 1, poté mohou jedinci neustále „kroužit“ kolem globálního optima, ale nikdy se do něj nedostanou.

Koeficienty r_1 a r_2 jsou náhodně vygenerovaná reálná čísla z intervalu $<0, 1>$. Koeficienty c_1 a c_2 jsou tzv. učící. Těmito koeficienty se určuje, jak rychle se budou jedinci pohybovat vzhledem nalezenému extrému. Pokud bude výrazně vyšší číslo koeficientu c_2 , který se nachází v členu (vzorec 2.3) u aktuálního nejlepšího řešení, poté je možná rychlá konvergence do lokálního extrému. Nikde není zaručeno, že první nalezený extrém je globální.

Jindy se místo c_1r_1 uvádí přímo $U(0, \varphi_1)$, což znamená náhodné číslo s uniformním rozložením. To samé platí pro c_2r_2 nahrazené $U(0, \varphi_2)$. Pokud je φ_2 větší než φ_1 (popsáno výše) směřují částice ke

² V základní verzi algoritmu. Pro lepší konvergenci algoritmu je možné hodnoty koeficientů měnit.

(možnému) globálnímu extrému. Tento jev se hodí v případě jedno-modálního problému. Naopak vyšší ϕ_1 je vhodné pro multi-modální problém [8].

Je nutné definovat, co se stane s jedinci, kteří opustí prostor. Je možné dané jedince dále neuvažovat nebo dané jedince vrátit na určitá místa, anebo náhodně vygenerovat jejich novou pozici v prostoru.

2.3.5 Ohodnocení jedinců

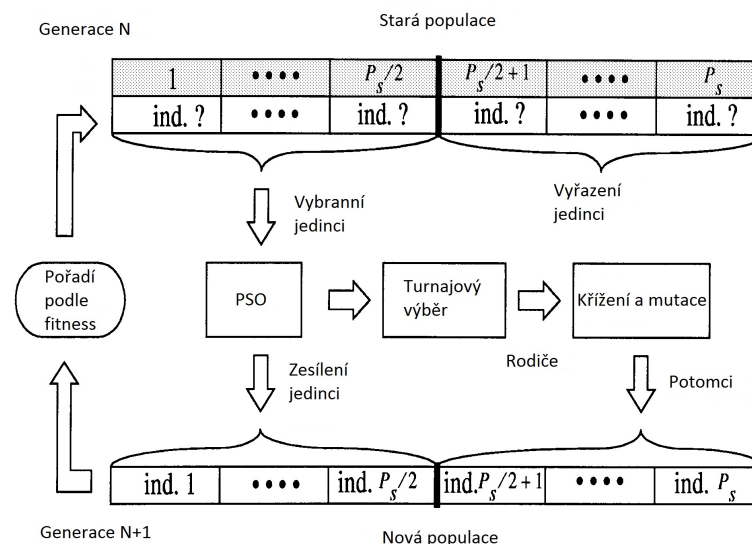
Podobně jako u evolučních případně genetických algoritmů [5] je zapotřebí jedince ohodnotit, aby byla zřejmá kvalita daného jedince. U PSO se rozumí spíše pozice jedince, která určuje hodnoty prohledávaného prostoru.

Ohodnocení opět záleží na zadaném problému, neexistuje přesný předpis pro všechny zadání. Stejně jako u genetických algoritmů při hledání globálního maxima budou kvalitnější jedinci s vyšším ohodnocením, v případě globálního minima s nižším.

Nevhodně navržená funkce ohodnocení maří snahu algoritmu o nalezení optimálního řešení. Jedinci by se pohybovali jiným směrem, než je globální extrém.

2.4 Hybridní algoritmus

Poslední zmíněnou technikou je spojení genetického algoritmu a optimalizace hejnem částic. Jedná se o hybridní algoritmus [24], který je znázorněn na obrázku 2.7. Zkráceně se označuje HGAPSO. Algoritmus neobsahuje nic co by nebylo již popsáno, ale zmíněné spojení by mohlo vést k lepším výsledkům.



Obrázek 2.7 HGAPSO [24]

Na začátku je nutná náhodná inicializace populace, poté dochází k ohodnocení a seřazení jedinců podle jejich kvality. V další části algoritmu je uvažována polovina lépe ohodnocených jedinců. Druhá polovina populace je zahozena.

Na vybrané jedince je aplikován operátor zesílení, což není nic jiného než algoritmus optimalizace hejnem částic. Výstup algoritmu představuje polovinu nové populace a také vstup genetického algoritmu. Na tyto jedince je aplikována turnajová selekce, která vrací rodičovskou populaci ke křížení a mutaci. Výsledkem těchto operátorů je druhá polovina nové populace.

Následně dochází k ohodnocení jedinců a seřazení podle jejich hodnoty fitness. Algoritmus je možno ukončit počtem generací nebo při dosažení dostatečně kvalitního řešení.

2.5 Shrnutí

Kapitola 2 ukázala dva druhy optimalizačních technik, které jsou inspirovány přírodou. První byly genetické algoritmy. U těchto algoritmů je velice důležité zvolit správnou formu zakódování řešeného problému. Také je velmi důležitá správně navržená fitness funkce, z důvodu usměrnění algoritmu správným směrem (globální optimum).

Druhou představenou technikou je Optimalizace hejnem částic (PSO), která je inspirována komplexním chováním skupiny, za kterým se skrývají jednoduchá pravidla jednotlivců. Na podobných principech existují i další techniky jako mravenčí kolonie a inteligence roje. Zmíněné techniky jsou inspirovány přírodou a mají jednu důležitou vlastnost. Všechny jsou schopné nalézt lepší řešení, než běžné metody optimalizace, ale také nemusí nalézt řešení vůbec. Respektive řešení nalezne vždy, ale může být horší oproti nalezeným řešením konvenčních metod optimalizace.

3 Nastavení vozu

Kapitola popisuje závodní vůz a jednotlivé aspekty, které je důležité zohlednit v nastavení jednotlivých parametrů vozu. Také je zmíněno rozdělení podle závodních vozů respektive závodních sérií, jednotlivých povrchů tratí atd. Následně je vypsán přehled vybraných parametrů a vysvětlení změn při různém nastavení. Ke konci kapitoly je zmíněn simulátor TORCS, především možnost parametrů, které je možné uvažovat. Kapitola vychází z [11], [12], [13], [14], [15], [16].

3.1 Obecný přehled

Nastavení parametrů závodního vozu je velice důležité. Pokud se vezmou v úvahu závodní série jako celek, jedná se o jeden velký koloběh finančních prostředků. Velice zjednodušeně řečeno pokud se vyhrává závod za závodem, jde největší podíl peněz od sponzorů, promotéra soutěže, z reklamy apod. k vítěznému týmu sezóny. Ekonomická situace týmu se odvíjí od těchto příjmů. Čím více peněz je získáno, tím více si může tým dovolit vynaložit náklady na vývoj vozu, testování komponent (z vývoje), platy vynikajících pilotů atd. Důležitou postavou v týmu, ohledně nastavení vozu, je hlavní závodní inženýr týmu, který má spolu s piloty hlavní slovo při volbě jednotlivých parametrů pro daný závod. Čím je závodní inženýr úspěšnější (zkušenější), tím roste jeho plat. Tyto faktory tvoří již zmíněný kolotoč peněz. Je totiž možné mít závodní vůz obsahující nejlepší komponenty na trhu, závodní vůz bude řídit skvělý jezdec, ale pokud bude nastavení vozu provedeno špatně, není možné zajet nejlepší čas.

Nedá se s určitostí říci, kolik špatných úsudků je možné si dovolit. Chyba v závodě, který rozhoduje o úspěchu případně neúspěchu celé sezóny, má za následek výrazný úbytek příjmů. Stabilizovaný tým tento fakt nepoloží, ale stává se nepříjemnou komplikací. Ekonomicky slabší tým může z důsledku podobné chyby zaniknout. Pokud je výsledek sezóny rozhodnutý, nemá špatný úsudek v nastavení až takový vliv. Ovšem i v tomto případě ztratí tým část svých příjmů.

3.2 Závodní informace

Nejdříve je nutné si představit aspekty jednotlivých druhů vozů, povrchů a důsledků které z toho vyplývají. Na začátek se představí vozy nejznámějších závodních sérií, poté povrchy tratí a následně se představí obecný model závodního víkendů.

3.2.1 Závodní série

V práci jsou uvažovány 3 nejznámější závodní série a to Formule 1 (dále F1) [17], závody rallye konkrétně WRC (World Rally Championship) [18] a závody cestovních vozů WTCC (World Touring Car Championship) [19]. Jednotlivé vozy jsou seřazeny na obrázcích 3.1 – 3.3. Vozy ostatních závodních sérií jsou vždy podobné některému zmíněnému vozu, respektive sdílí podobné nastavení parametrů.



Obrázek 3.1 Vůz F1
[27]



Obrázek 3.2 Vůz WRC
[28]



Obrázek 3.3 Vůz
WTCC [29]

3.2.2 Povrch trati

Důležitý fakt, který je nutné vzít v úvahu především v rámci obutí jednotlivých vozů. Spektrum povrchů trati je různé. Pro závody F1 a cestovní vozy WTCC je spektrum tvořeno různými druhy asfaltu (především drsnosti). Také je nutné počítat s nepříjemnostmi městských okruhů, jako jsou bílé čáry (vodorovné dopravní značky), poklopy kanalizace apod. I když toto je již úkol spíše pilota na trati. V případě závodů rallye je spektrum povrchů trati pestřejší přes asfalt, šotolinu až po sníh a led.

3.2.3 Formát závodního víkendu

Závodní víkend probíhá v každé sérii (mírně) odlišně, ale pro potřeby práce je možné sjednotit všechny série. V krátkosti jsou představeny 3 uvažované závodní série.

U závodů F1 [17] se začíná pátečními tréninky, které dávají velkou možnost nastavení celého vozu. V sobotu dopoledne je možnost nastavení opět v tréninku, ale zde už se jedná o detaily. Odpoledne se jede kvalifikační trénink a na základě časů z tohoto tréninku se piloti kvalifikují do závodu. V neděli se jede hlavní závod.

Pro závody WTCC [19] je formát víkendu podobný jako u F1, pouze závodní víkend je smrštěný do soboty a neděle. V případě této série se jedou v neděli dva kratší závody.

Odlišněji působí závody rallye [18]. Už z principu se závody liší soubojem proti času, než přímým soubojem vozů na trati. Také se nejedou okruhové závody, ale etapové. Nastavení vozu probíhá před začátkem každé etapy.

3.2.4 Prvotní nastavení

Tato část se týká především závodů F1 a WTCC. Prvotní nastavení [12] probíhá (před prvními tréninky) ze znalosti okruhů z předchozích let nebo pro různé parametry se zvolí neutrální nastavení podle specifikace okruhu. Poté první tréninky probíhají způsobem obkroužení jednoho okruhu a zpět do garáže. Velice důležité je změny dělat postupně v co nejmenším počtu. Poté opět zajet jedno kolo na okruhu (případně více). Je nutné přesně stanovit vliv změny parametru, které se projeví změnou chování vozu. Změna se projeví pozitivně, negativně nebo se neprojeví vůbec. Vše je důležité sledovat pro usměrnění správného nastavení vozu.

3.3 Parametry závodního vozu

Na začátek je důležité říci, že změny parametrů vozu probíhají z důvodu usměrňování sil [12], které se projevují ve vysokých rychlostech. Především přítláčná síla je velice důležitá.

V případě akcelerace je snaha usměrnit sílu, která působí na zadní část vozu. Čím se zvyšuje rychlost, tím je menší akcelerace. Čili na konci rovinných úseků, kde je snaha dosažení co největší rychlosti, se akcelerace blíží k nule. Na těchto úsecích je zapotřebí co nejmenší přítláčné síly.

Opačným případem je brzdění, kdy se váha vozu přesunuje na jeho přední část. Brzdění probíhá, pokud je vůz stále na rovinném úseku a blíží se zatáčka. Je nutné skutečně brzdit na rovince. Brzdění v zatáčce by způsobilo přetáčivost případně nedotáčivost vozu (viz kapitola 3.3.1 Brzdy). Další fáze je průjezd zatáčkou, kdy se váha přenáší na vnější část vozu. V případě levotočivé zatáčky je pravá část vozu vnější (analogicky pro pravotočivou zatáčku). Při průjezdu zatáčkou je důležitá kontrola vozu. V této části je naopak zapotřebí co největší přítláčné síly. Na výjezdu ze zatáčky začíná vůz akcelerovat (viz předchozí odstavec).

V podkapitolách se často odkazuje na profil okruhu. K objasnění byly vybrány dva okruhy, které jsou svým profilem naprosto odlišné. Na obrázku 3.4 je vyobrazený okruh v Monze (Itálie). Okruh se vyznačuje vysokou rychlostí (průměrnou i maximální). Ihned po delších rovinných úsecích se nachází ostré zatáčky, případně rychlá kombinace ostrých zatáček tzv. „šikan.“ Na obrázku 3.5 je městský okruh nacházející se v ulicích Monte Carla (Monako). Okruh je velice pomalý, čili se nedosahuje vysokých rychlostí. I když jsou některé zatáčky ostřejší, v nastavení se tento fakt neprojeví tolik jako na rychlejším okruhu. Z obrázků to sice není patrné, ale délky okruhu se výrazně liší. Okruh v Monaku má délku okruhu 3,3 km. Italský okruh má délku 5,8 km.

Nyní již k jednotlivým nastavitelným parametrům závodního vozu. Záměrně byl vynechán motor, jelikož motory jsou omezeny pravidly jednotlivých sérií a mnoho parametrů je neměnných. Například u F1 je vývoj motoru nepřipustný (vývoj motorů je tzv. zmrazen).



Obrázek 3.4 Okruh v Monze [26]



Obrázek 3.5 Okruh v Monaku [25]

3.3.1 Brzdy

Brzdy [12] umožňují nastavení brzdného tlaku, vyvážení brzd a průměrů brzdných kotoučů. Nastavení brzdného tlaku reguluje zablokování kol při sešlápnutí brzdného pedálu. Při zablokování kol se dostává vůz do smyku a stává se neovladatelným. Brzdný tlak se volí na základě profilu trati. Pokud je profil podobný obrázku 3.4 je nutné nastavit brzdný tlak větší. Také průměr brzdných kotoučů bude větší. Naopak při profilu trati podobnému jako obrázku 3.5 se volí brzdný tlak a průměr brzdných kotoučů menší.

Nastavení vyvážení je poměr mezi předními a zadními brzdami (udává se v procentech). Ideální nastavení by bylo 50% na předních brzdách a 50% na zadních brzdách. Při brzdění se přenáší váha vozu na přední část vozu. Tento fakt je potřeba zohlednit při nastavení vyvážení. V mnoha případech se volí hodnoty cca 60% na předních a 40% na zadních brzdách. Nastavení na předních brzdách by nemělo překročit 70%. Poté se stává vůz přetáčivým při nájezdu do zatáček. Pokud by bylo na zadních brzdách více než 50%, vůz by se stal nedotáčivým (nedostal by se správně na střed zatáčky).

3.3.2 Pružiny

U pružin [13] se nastavuje jejich délka a tuhost. Největší interval hodnot pro toto nastavení bude při závodech rally. Důvodem je pestrost povrchů jednotlivých etap. Nastavení delších a měkčích pružin se volí při nerovném povrchu jako je například šotolina. Toto nastavení způsobí, že kola budou lépe kopírovat nerovnosti terénu tj. nebudou po nerovnostech odskakovat. Naopak nastavení kratších a tvrdších pružin je voleno pro asfaltový povrch, kde nerovností ubývá a vůz bude lépe ovladatelný při tomto nastavení.

3.3.3 Tlumiče

U tlumičů [13] se nastavuje tuhost (podobně jako u pružin). Jejich úkolem je tlumit nárazy, které vznikají působením sil v jednotlivých fázích okruhu. Vše je způsobeno přesunem váhy vozu v určitých momentech.

Při brzdění přední tlumiče tlumí náraz a zadní odskok. Při průjezdu levotočivou zatáčkou tlumiče na levé straně vozu tlumí odskok (tlumiče se natahují) a na pravé straně náraz (tlumiče se stlačují). Analogicky pro pravotočivou zatáčku. Na výjezdu ze zatáčky (při akceleraci) přední tlumiče tlumí odskok a zadní náraz.

Nastavení tuhosti má za následek rychlost natahování případně vytahování tlumičů. Tímto nastavením se ovládá přesun váhy celého vozu. Čím měkčí je tlumič, tím se váha vozu rychleji přesouvá a naopak. Nastavením měkčího tlumiče je možno vyřešit nedotáčivost vozu. Ovšem pokud se nastaví přední tlumič až moc měkký, poté bude mít vůz špatné vlastnosti při akceleraci. Nedotáčivost vozu by se proto měla tlumiči pouze doladit.

3.3.4 Příčné stabilizátory

U tohoto parametru [11] opět nastavujeme tuhost. Příčné stabilizátory se snaží zabránit náklonu vozu v zatáčkách. Nacházejí se na obou nápravách vozu. Čím bude příčný stabilizátor tužší, tím lépe bude vůz sedět v zatáčce. Bude se také lépe ovládat, protože kola budou mít stálý kontakt s vozovkou. Ovšem s tímto nastavením je možné počítat pouze při asfaltovém povrchu. Čím bude povrch mít více nerovností, tím je potřeba nastavit měkčí stabilizátor.

Pokud je zapotřebí nastavit vůz (například nedotáčivost) na celý okruh, je vhodné nastavovat stabilizátory. Nastavení vozu z důvodu jedné specifické zatáčky, si žádá nastavit tlumiče.

3.3.5 Světlá výška

Světlá výška [11] je vzdálenost mezi podvozkem a vozovkou. Parametr se nastavuje v závislosti na nerovnostech okruhu. Pokud je povrchem asfalt, je vhodné snížit těžiště vozu z důvodu lepší ovladatelnosti. Toho se dosáhne snížením světlé výšky vozu. V případě nerovného povrchu (šotolina) je zapotřebí světlou výšku zvýšit. Avšak zbytečně vysoká světlá výška znamená horší ovladatelnost vozu. Naopak vůz s velmi nízkou světlou výškou způsobí třením o vozovku výrazné zpomalení vozu. V horším případě může dojít k poškození vozu. Tato fakta platí pro obě varianty povrchu (asfalt, šotolina).

3.3.6 Podvozek

Poslední 4 zmíněné části vozu (Pružiny, Tlumiče, Stabilizátory a Světlá výška) nastavují podvozek vozu. Komplexní nastavení těchto parametrů ovlivňuje ovladatelnost celého vozu. Například nastavení menší světlé výšky umožňuje změnit tuhost stabilizátorů. Změna parametrů podvozku je nutná i v případě nastavení ostatních částí vozu. Například změnou vyvážení brzd bude pravděpodobně nutné změnit nastavení celého podvozku. V opačném případě by se mohla projevit nedotáčivost (případně přetáčivost) vozu.

3.3.7 Přítlačná křídla a difuzor

Nastavení přítlačných křídel [12] se týká především F1, kde se nastavuje úhel natočení křídla. Přítlačná křídla určují přítlačnou sílu vozu. Obrázek 3.4 ukazuje okruh, který je tvořen dlouhými rovnými úseky. Zde není zapotřebí velké přítlačné síly. Přítlačná křídla se v tomto případě nastaví téměř do roviny. Pokud se jedná o okruh na obrázku 3.5, kde je nutná větší přítlačná síla, křídla se natočí pod větším úhlem. Při změně natočení předního křídla se musí změnit i natočení zadního.

Difuzor se nachází na spodní straně vozu a ústí v jeho zadní části. Usměrněje vzduch, který proudí pod vozem. Správné nastavení světlé výšky, předního přítlačného křídla a difuzoru vyvíří podtlak mezi podlahou vozu a vozovkou. Toto nastavení výrazně ovlivňuje výslednou přítlačnou sílu.

3.3.8 Převodovka

Převodovka udržuje motor v optimálních otáčkách. Základním parametrem je počet převodových poměrů. U závodních vozů se volí 6 nebo 7 (plus zpětný chod). Menší počet by nedovolil vozu dostat se na svoji maximální rychlost. Vyšší počet by vedl k velkým časovým ztrátám při přefazování.

Jednotlivé převodové poměry se nastavují svojí délkou [13]. Čím je převodový poměr vyšší, tím se převod zkracuje. První se nastavuje nejvyšší převodový poměr v závislosti na maximální rychlosti okruhu. Nastaví se tak, aby maximální rychlost odpovídala otáčkám, při kterých má motor maximální výkon. Po nastavení nejvyššího převodového stupně se pokračuje převodem nejnižším. První stupeň je využit z 90% na rozjezd, který musí být v ideálním případě plynulý a bez proklouznutí kol. Proto se nastaví na nejdelší možnou délku, dokud motor neztrácí výkon. Následující převodové stupně se nastavují kratší a kratší, jelikož při přefazení na vyšší rychlostní stupně, klesají otáčky méně a méně.

Nastavení převodových poměrů se také odvíjí od profilu okruhu. Okruh vyobrazený na obrázku 3.4, na kterém se dosahuje několikrát maximálních rychlostí, dovoluje nastavení delších převodových poměrů. Naopak okruh na obrázku 3.5, na kterém se maximální rychlosti vozu nedosáhne, si

vyžaduje kratší převodové stupně. Na tomto okruhu je také nutná velmi dobrá akcelerace ze zatáček (kratší převodové stupně).

3.3.9 Rozvodovka a diferenciál

Rozvodovka umožňuje rozvod kroutícího momentu z jedné hřídele na dvě (i více) hřídele. U (závodních) vozů se přenáší kroutící moment z hřídele převodovky na nápravu. Součástí rozvodovky je diferenciál.

Existují různé typy diferenciálů [13]. Jejich funkce spočívá v umožnění prokluzu kol při zatáčení. Vnitřní kola opisují menší dráhu než vnější, tudíž vznikají různé otáčky kol.

Prvním druhem diferenciálu je uzavřený diferenciál. Pro závodní série zmiňované v této práci je nepoužitelný, jelikož se nehodí závodní tratě se zatáčkami. Diferenciál nedovoluje prokluz a kola se točí stejnou rychlostí. Čili tento druh diferenciálu se dá označit za výjimku.

Druhým typem diferenciálu je diferenciál otevřený. Tento diferenciál umožňuje rozdílnou rychlost kol (dovolen prokluz). Pro uvažované závodní série má lepší vlastnosti než diferenciál uzavřený.

Třetím typem a zároveň posledním zde uvedeným, je samosvorný diferenciál. Využívá lamelových spojek, které rozpojí hřídel nápravy (od rozvodovky ke kolu) v pravý moment. Tímto se zajistí rozdílná rychlost kol a maximální možná akcelerace.

3.3.10 Maximální rejď

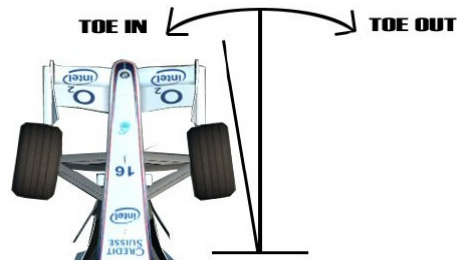
Změnou nastavení maximálního rejdu [12], se změní úhel natočení kol při maximálním otočení volantu. Maximální rejď se nastavuje v závislosti na délce rozvoru náprav. Čím větší je rozvor náprav, tím větší musí být maximální rejď k dosažení stejného efektu při zatáčení. Větší maximální rejď způsobuje citlivější, ale i méně přesné řízení. Opačné vlastnosti platí pro nastavení menšího maximálního rejdu.

3.3.11 Sbíhavost kol

Sbíhavost [13] znázorňuje obrázek 3.6. Při pohledu shora se jedná o natočení přední části kol k sobě. Nulová sbíhavost je, pokud jsou kola rovnoběžná. Pokud jsou přední části kol natočeny od sebe, mluví se o tzv. rozbíhavosti.

U předních kol se nejčastěji volí mírná rozbíhavost, z důvodu správného natočení kol do zatáček. Na rovinném úseku vůz nepojede přesně rovným směrem, zde se ovšem dá spolehnout na šikvost pilota. V opačném případě (sbíhavosti) by se vůz choval velice dobře na rovinných úsecích, ale měl by špatné vlastnosti v zatáčkách (horší varianta). U zadních kol je velice často volena mírná

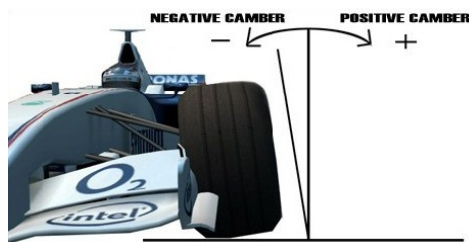
sbíhavost. Rozbíhavost by byla volena pouze v případě náhonu na přední kola. Ovšem tato varianta není u závodních vozů běžná.



Obrázek 3.6 Sbíhavost kol [13]

3.3.12 Odklon kol

Na obrázku 3.7 je znázorněn odklon kol [13]. Při pohledu ze předu je negativním odklonem myšlen náklon horní části kol k sobě. Nulový náklon znamená, že kola jsou rovnoběžná. Pozitivní odklon je natočení kol horní částí od sebe.



Obrázek 3.7 Odklon kol [13]

Odklon kol má velký význam na pneumatiky (zahřátí a opotřebení). Nezáleží tolik na pozitivním nebo negativním odklonu, ale na velikosti úhlu. Na rovinných úsecích by byly zapotřebí větší úhly, jelikož bude mít pneumatika menší odpor a dosáhne dříve maximální (případně větší) rychlosti. Ovšem v zatáčkách je zapotřebí co největší kontakt s vozovkou (lepší ovladatelnost) čili menší úhel odklonu. Při menším odklonu dochází k rovnoměrnému opotřebení i lepšímu zahřátí pneumatik. Z tohoto důvodu se na obou nápravách volí mírné odklony kol.

3.3.13 Rozchod kol

Nastavení rozchodu kol [13], není nic jiného než změna vzdálenosti mezi levým a pravým kolem, jak na přední tak i na zadní nápravě. Čím větší rozchod kol bude, tím bude větší přilnavost. Ovšem při nastavení velkého rozchodu kol, bude vůz širší a také kola (při pohledu ze shora) budou „do čtverce.“ Poté bude horší stabilita vozu v přímém směru.

3.3.14 Geometrie kol

Poslední 4 podkapitoly (Maximální rejď, Sbíhavost kol, Odklon kol a Rozchod kol) souvisí s nastavením správné geometrie kol. Je to podobné jako u nastavení podvozku. Při změně jednoho parametru je velice často nutné změnit další.

3.3.15 Pneumatiky

Pneumatiky jsou velmi podceňovanou částí závodního vozu. Pneumatika je jediný prvek, který je v kontaktu s vozovkou. Správné nastavení, výbava vozu, šikovnost pilota atd. může přijít nazmar z důvodu špatně zvoleného „obutí.“

Volí se vzorek, směs a tlak pneumatiky. Vzorek pneumatiky je závislý na klimatických podmínkách. Opotřebenění pneumatik je závislé na směsi. Měkčí pneumatiky se opotřebovávají rychleji, ale mívají lepší jízdní vlastnosti. Vlastnosti tvrdších pneumatik jsou opačné.

Volbou správného tlaku [12] v pneumatikách lze dosáhnout lepších vlastností ovladatelnosti a opotřebenění pneumatik. Každá pneumatika má dané rozmezí (interval), ve kterém je schopna podávat nejlepší výkony. Nastavení přesného tlaku závisí na volbě každém týmu. V případě nižšího tlaku se pneumatika lépe zahřívá a opotřebenění je rovnoměrné po celé její ploše. Také v zatáčce má pneumatika lepší přilnavost. Naopak při vyšším tlaku dochází k menšímu opotřebenění pneumatiky a dosahuje vyšších rychlostí na rovinných úsecích. Ovšem opotřebenění nemusí být rovnoměrné. Jedna strana se může opotřebovávat více než druhá (záleží na odklonu kol). Při vyšším tlaku se pneumatika hůře zahřívá. Správné a rychlé zahřátí pneumatiky vede k lepší přilnavosti vozu.

Volba prvních dvou parametrů záleží spíše na vnějších okolnostech. Volba tlaku v pneumatikách záleží na konkrétním pilotovi a jeho jízdním stylu.

3.4 Simulátor TORCS

V rámci kapitoly bude představen simulátor TORCS (The Open Racing Car Simulator) se zaměřením na princip a architekturu. Jedná se o simulátor závodních vozů, který je volně dostupný. Další aspekty jako například instalace, ovládání, jsou předmětem uvedení programu do chodu jsou uvedeny v kapitole 7. Kapitola vychází z [14] a [15].

3.4.1 Popis programu

Simulátor TORCS [14] je stand-alone³ aplikace, ve které jsou samostatné části (bots) zkompileovány jako samostatné moduly, které jsou po dobu závodění (používání programu) nahrány

3 Program běžící jako samostatný proces bez nutnosti síťového připojení

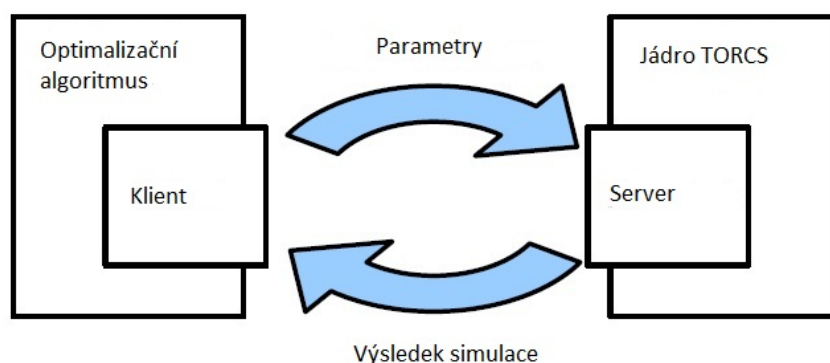
do operační paměti. Tyto části umožňují nastavovat parametry, které je potřeba modifikovat v rámci optimalizace při řešení problémů s nastavením vozu. Což je dobrý základ pro optimalizační algoritmy. Simulátor TORCS má také nevýhody.

První nevýhoda spočívá v tom, že nejsou odděleny programovatelné části od simulačního jádra (engine). Jednotlivé části mají přístup ke všem datům reprezentujících auto i trať. Tím pádem různé optimalizační techniky s různým přístupem, se mohou zaměřit na veškeré parametry s využitím přístupu k herním strukturám, aby dosáhly co nejlepších výsledků. Což je samozřejmě jejich účel, ale poté je těžké jednotlivé přístupy porovnat. Různé přístupy se mohou zaměřit na potřebné parametry (mají znalost, kterou by mít neměly).

Druhou nevýhodou je omezení v rámci architektury TORCS, kdy je nutné použít existujících optimalizačních struktur (frameworks) a také omezení v rámci programovacího jazyka na C++. Simulátor TORCS je napsaný právě v tomto programovacím jazyce.

3.4.2 Rozšíření TORCS

Existuje rozšíření [15] původní architektury TORCS, které má v sobě několik vylepšení. První vylepšení je, že aplikace TORCS je nyní typu klient server. Jednotlivé části běží jako vnější procesy připojené k serveru. Komunikace probíhá pomocí protokolu UDP.



Obrázek 3.8 Architektura softwaru pro optimalizaci [15]

Druhé vylepšení spočívá v možnosti měnit parametry v průběhu závodu, dříve byly načteny na začátku závodu. Posledním vylepšením je oddělení optimalizace od serveru, vytvořením abstraktní vrstvy. Tato vrstva dává úplnou volnost výběru programovacího jazyka pro optimalizační techniky nezávislé na programovacím jazyce, použitým v jednotlivých částech simulátoru. Také omezuje přístup k těm parametrům, které definuje návrhář (programátor) a omezuje jednotlivé přístupy, aby se nevyužívaly herní struktury (řešení problému přístupu ke všem datům). Architektura celého procesu optimalizace je zobrazena na obrázku 3.8. Zde je vidět rozšíření především v části serveru (server

bot), která řídí komunikaci mezi hrou a klientem, kde běží optimalizační metody, využívající protokol UDP.

3.4.3 Nastavitelné parametry vozu v TORCS

V simulátoru TORCS je možné nastavit 22 parametrů. Jsou seřazené v tabulce 3.1, kde jsou uvedené hodnoty, kterých jednotlivé parametry mohou dosahovat.

Parametr	Díl	Název	Jednotka	Min	Max
1	Převodovka – převod 2	poměr	SI	0	5
2	Převodovka – převod 3	poměr	SI	0	5
3	Převodovka – převod 4	poměr	SI	0	5
4	Převodovka – převod 5	poměr	SI	0	5
5	Převodovka – převod 6	poměr	SI	0	5
6	Zadní křídlo	úhel	stupeň	0	18
7	Přední křídlo	úhel	stupeň	0	12
8	Brzdy	poměr přední-zadní	SI	0,3	0,7
9	Brzdy	maximální tlak	kPa	100	150000
10	Přední stabilizátor	pružina	libra/palec	0	5000
11	Zadní stabilizátor	pružina	libra/palec	0	5000
12	Přední kola	světla výška	mm	100	300
13	Přední kola	sbíhavost	stupeň	-5	5
14	Přední kola	odklon	stupeň	-5	3
15	Zadní kola	světla výška	mm	100	300
16	Zadní kola	odklon	stupeň	-5	-2
17	Přední zavěšení	tlumič	libra/palec	0	10000
18	Přední zavěšení	výška	m	0	0,2
19	Zadní zavěšení	tlumič	libra/palec	0	10000
20	Zadní zavěšení	výška	m	0	0,2
21	Přední brzdy	průměr kotouče	mm	100	380
22	Zadní brzdy	průměr kotouče	mm	100	380

Tabulka 3.1 Parametry vozu v simulátoru TORCS [15]

3.4.4 Popis komunikace

Prvním krokem je inicializace [14] během níž se klient připojí na server a získá zpět dvě hodnoty:

- čas simulace
- čísla parametrů.

Čas simulace označuje, jak dlouho bude ještě celková simulace trvat. Seznam parametrů může dodržovat pořadí uvedené v tabulce 3.1 nebo může využít náhodné mapování z důvodu vyloučení předchozí znalosti. Přestože jsou v tabulce 3.1 uvedeny rozdílné rozsahy hodnot u jednotlivých parametrů, je uvnitř optimalizačního procesu (klient) předpokládán rozsah od 0 do 1. Server si pokaždé za pomoci mapování převede z rozsahu 0-1 do skutečného. Pokud server správně identifikuje klienta, začíná celý proces. Auto (proces serveru) bude zablokované, dokud od klienta nedorazí žádost o vyhodnocení.

Po inicializaci se spustí optimalizační algoritmus. Pokud je zapotřebí vyhodnocení, dojde k těmto krokům [15]:

- Klient pošle seznam parametrů na server. Seznam musí mít stejnou velikost jako číslo určující počet parametrů a všechny hodnoty musí být v rozsahu od 0 do 1. Společně s těmito informacemi posílá klient hodnotu odpovídající simulačnímu času.
- Server přijme tyto hodnoty, proběhne namapování hodnot do skutečných rozsahů parametrů vozu a načte parametry v simulačním jádře (engine).
- Vůz startuje jízdu podle daného scénáře.
- Po vypršení simulačního času vyhodnocení skončí a vůz zastaví.
- Server pošle zpět výsledky vyhodnocení (uraženou dráhu, maximální rychlost, poškození, nejlepší čas na kolo).
- Klient použije tyto hodnoty k usměrnění optimalizačního algoritmu.

Každé vyhodnocení bývá provedeno po uplynutí času simulace, což je nastavitelná část z přiděleného času celkové simulace. Tudiž je důležité správně nastavit tento čas. Pokud bude čas příliš krátký, vůz pojede pouze menší počet kol, což vede ke špatnému ohodnocení. Při extrémně malé hodnotě nemusí vůbec dokončit kolo. V tomto případě vrácené hodnoty nebudou mít vypovídající hodnotu (nejlepší dosažený čas na kolo bude mít hodnotu -1).

Naopak při vysoké hodnotě simulačního času je možné, že vyprší celkový čas simulace a optimalizační algoritmus proběhne ve velmi malém počtu iterací. Neboli server bude neustále zpracovávat jednu sadu nastavení.

Pokud se přeruší simulace, z důvodu vyhodnocení aktuálních výsledků, další běh začne v bodě, kde předchozí simulace skončila. Což se může projevit ve výrazně odlišných jednorázových výsledcích.

Posledním krokem je ukončení optimalizace [15]. V případě vypršení celkového času simulace, server již nic dalšího neprovádí a pošle klientovi zprávu o vypršení času. Klient v tu chvíli musí odeslat nejlepší dosažený výsledek.

Klient velice často monitoruje hodnotu určující zbývající čas simulace (sumou časů předešlých ohodnocení). Tímto předchází nedokončení ohodnocení aktuálního nastavení, kdy už nezbývá přidělený čas procesoru.

3.5 Shrnutí

V kapitole Nastavení vozu byly uvedeny nejdůležitější parametry závodního vozu. Bylo snahou popsat jednotlivé parametry v pořadí, které se podobá reálnému postupu. Ovšem každý tým má svoje vlastní postupy, tudíž tento postup nelze popsat zcela přesně. Ke konci kapitoly byl představen simulátor TORCS.

4 Návrh řešení

Kapitola představuje návrhy realizací zvolených optimalizačních technik. Jsou zde použity optimalizační techniky představené v kapitole 2. Na začátku kapitoly je představen projekt CIG [21]. Poté je naznačeno ohodnocování jedinců, jelikož princip ohodnocení je podobný pro každou optimalizační techniku.

4.1 Představení projektu CIG

Projekt CIG (Computational Intelligence in Games) [21] respektive jeho druhá varianta (hledání nejlepšího nastavení vozu) probíhá na 3 zvolených okruzích. Optimalizační algoritmus hledá nejlepší možné nastavení pro každý okruh.

Ve fázi optimalizace pracuje algoritmus na hledání nejlepšího řešení. Ve fázi ohodnocení je na serveru spuštěn proces ohodnocení. V tyto momenty se přičítá čas procesoru (serverová část). Limit je pro všechny nastaven na 2 hodiny času procesoru. Po uplynutí tohoto času zašle server oznámení s chybou. Po (dokončené) fázi ohodnocení server vrátí hodnotu nejlepšího dosaženého času na kolo, maximální rychlosti, odjeté vzdálenosti a míru poškození vozu. Na základě těchto hodnot dojde k ohodnocení jednotlivých řešení.

4.2 Ohodnocení výsledků

Již podle kapitoly 2 Optimalizační techniky je velice důležité správně navrhnout funkci, která ohodnotí aktuální jedince. V rámci všech technik je možné tuto funkci označit jako fitness.

Ohledně problému nastavení závodního vozu je důležité zohlednit několik hledisek. Výstupem serverové části (3.4 Simulátor TORCS) jsou 4 hodnoty: nejlepší čas na kolo, ujetá vzdálenost, maximální rychlost a míra poškození vozu. Nejdůležitější hledisko pro ohodnocení daného jedince je nejlepší čas na kolo. Čas na kolo většiny okruhů se pohybuje okolo minuty a půl. Jeho formát je mm:ss:hhh (m – minuty, s – sekundy, hhh – tisíciny sekundy). K jednotlivým ohodnocením jedinců je nutné započítat zlepšení o každou tisícinu sekundy. V motoristickém sportu může rozhodovat i menší počet setin. Posun času o desetinu sekundy na kolo je již posunem výrazným. Pro každé ohodnocení je tedy započítáno zlepšení o každou tisícinu sekundy. Tento fakt je důležitý především ke konci běhu algoritmu, kdy algoritmus jakékoliv optimalizační techniky bude pomaleji konvergovat.

Ovšem v rámci správně navržené fitness funkce je důležité přihlídnout i k dalšímu hledisku. Především je třeba zohlednit míru poškození, jelikož vůz může zajet vynikající kolo, ale míra

opotřebením součástí bude tak velká, že nebude schopen dále pokračovat. I když k tomuto problému je možné přistoupit dvěma způsoby:

- 1) Poškození nebrat v úvahu. Projekt CIG je hledání nejlepšího nastavení na jedno kolo zvoleného okruhu.
- 2) Poškození zohlednit, jelikož nepřichází v úvahu žádný závod, kde by se uplatnilo nastavení vozu pouze na jedno kolo. Výjimku by mohl tvořit kvalifikační trénink, ale i jeho forma již neuvažuje zjetí pouhého (jednoho) kola. Tento fakt platí pro drtivou většinu závodních sérií.

K výstupu práce budou uvažovány obě varianty, kde bude zajímavé porovnat vliv poškození nebo lépe řečeno porovnání spolehlivosti jednotlivých komponent vozu za cenu horšího času na kolo. Horší čas na kolo je logickým předpokladem, vždy se musí hledat kompromis mezi rychlostí a spolehlivostí.

Další hledisko pro určení objektivní fitness hodnoty je ujetá vzdálenost vozu. Bylo by dobré stanovit určitý počet okruhů (okruh má vždy uvedenou svou vzdálenost), který musí vůz odjet. Za předpokladu startu z klidové polohy před startovní čarou je nutné obkroužit minimálně jedno kolo navíc. Následující kolo bude výrazně rychlejší z důvodu projetí startovní čáry v plné rychlosti. Započítání hlediska odjeté vzdálenosti je nutné také z důvodů vyhodnocení na straně serveru, kde vůz zastaví po vypršení určené doby k vyhodnocení aktuálního nastavení. Další vyhodnocení proběhne od místa, kde vůz zastavil. K odstranění nepřesností by mělo dojít po odjetí minimálně 3 kol. Tento parametr se nastavuje nepřímo v rámci zvolení času jednotlivých fází z celkového přiděleného času.

Poslední hledisko maximální rychlost nebude s velkou pravděpodobností bráno v úvahu. Výsledné ohodnocení by mohla maximální rychlost nevědomě ovlivnit v negativním smyslu. V případě okruhu se startovní čarou⁴ nacházející se u konce závěrečného rovinného úseku okruhu v blízkosti první zatáčky, by mohlo dojít ke špatnému brzdění z důvodu vysoké maximální rychlosti. Jinak řečeno čas na kolo by byl vynikající, ale s ohledem na příští kolo by nebylo nastavení moc dobré. Vůz by v první zatáčce vyjel mimo trať nebo v horším případě by skončil v bariéře.

V zásadě samotná fitness funkce není náročná na vytvoření, jelikož veškeré ohodnocení je provedeno na serverové části. Určení správné fitness hodnoty se týká použití správných koeficientů, kterými se násobí návratové hodnoty serveru.

4 Na určitých tratích jsou startovní a cílová čára od sebe vzdáleny desítky metrů, zde se uvádí případ, kdy jsou obě čáry na shodném místě

4.3 Evoluční algoritmus

V návrhu parametrů evolučního algoritmu (konkrétně genetického) budou popsány zásadní aspekty, které byly popsány v kapitole 2 Optimalizační techniky. Výjimku tvoří fitness funkce, která byla popsána v kapitole 4.2 Ohodnocení výsledků.

4.3.1 Zakódování problému nastavení vozu

Zadání problému ukazuje potřebu reálných čísel pro pokrytí rozsahu parametrů (Tabulka 3.1). Vzhledem k uvedeným rozmezím u žádného parametru není nutná větší přesnost než na 3 desetinná místa, která je více než dostačující.

Samotné zakódování je voleno odděleně pro každý parametr závodního vozu. Rovněž jsou odděleni jednotliví jedinci. Co se týče reprezentace samotných čísel, je volena v reálných číslech. Veškeré parametry závodního vozu jsou uloženy za sebou v daném pořadí. Intervalová omezení jednotlivých parametrů jsou brány v úvahu při prvotním náhodném vygenerování hodnot.

4.3.2 Počet jedinců a iterací algoritmu

Vzhledem k obtížnosti problému a také neustálému vyhodnocení aktuálních nastavení na serveru, je nutné zvolit vyšší počet jedinců v populaci. Jejich počet je navržen na 200 jedinců. V případě nutnosti by se dalo uvažovat i o 300 jedincích.

Zvolení počtu iterací je závislý na nastavení parametru času, který zabírá jedno vyhodnocení na serveru. Tento fakt bude ovlivňovat i počet jedinců v populaci. Počet iterací se pohybuje okolo sto tisíc.

4.3.3 Selektce, křížení, mutace a obnova populace

Nyní již ke zvolení jednotlivých parametrů genetického algoritmu. Co se týká selektce, zde je vybrána turnajová selektce, která má velmi dobré vlastnosti na správnou konvergenci algoritmu. Počet vybraných jedinců (selekcí) se pohybuje okolo 30-50% populace.

Křížení je možno aplikovat dvěma způsoby:

- 1) V případě reálného zakódování (jak je popsáno v kapitole 4.3.1) je nejjednodušší aplikovat křížení uniformní. Při tomto druhu křížení se vždy vybere parametr na aktuální pozici, čili není nutné řešit velké rozmezí veškerých parametrů.
- 2) Při použití vícebodového křížení bude nutná transformace reálného zakódování například do binárního řetězce, provést operátor křížení (případně i mutace) a zpětně transformovat do reálného zakódování. Počet bitů jednotlivých parametrů by byl předem daný, čili nemělo by dojít k opuštění rozsahu jakéhokoliv parametru.

V případě binární transformace se naskýtají dvě možnosti:

- a) Aplikovat křížení v rámci celkového řetězce bez ohledu na rozdělení parametrů.
- b) Využít operátor křížení na jednotlivé parametry odděleně.

Možnost a) je zcela jistě jednodušší, ale možnost b) by měla vnést do populace větší rozmanitost. U možnosti b) by mohlo být prospěšné před aplikací křížení na každý parametr vygenerovat nové náhodné číslo určující provedení operátoru křížení. Tento fakt by mohl vést na již zmíněnou rozmanitost populace.

Operátor mutace bude na začátku volen s pravděpodobností 0,1. V případě možnosti 1) (popsané u křížení) se bude jednat o přičtení (odečtení) náhodné hodnoty. U druhé možnosti bude aplikace mutace jednodušší, a sice provedení inverze bitů. I zde je možné provést mutaci podobně, jak jsou popsány možnosti a) a b) u křížení.

Po operátorech genetického algoritmu následuje ohodnocení (kapitola 4.2). Před zavoláním fitness funkce bude nutné převést čísla parametrů do rozsahů intervalu $<0, 1>$ (vyžaduje serverová část). Zde je zapotřebí přesnost na co nejvíce desetinných míst.

Obnova populace bude volena turnajem, náhradou cca 30% nové populace za nejhorší jedince populace předchozí.

4.4 Optimalizace hejnem částic

Kapitola obsahuje návrh veškerých parametrů, nutných k realizaci algoritmu PSO. Výjimkou je funkce pro ohodnocení jedinců, která je popsána v kapitole 4.2. Podobně jako u genetického algoritmu jsou hodnoty uvedeny v původním rozsahu a před vyhodnocením jsou převedeny do požadovaného rozsahu serveru (reálné číslo od 0 do 1).

4.4.1 Prohledávaný prostor, počet jedinců a komunikace

Prohledávaný prostor je určen počtem dimenzí. Každá dimenze představuje jeden parametr závodního vozu. Pokud bude uvažováno všech 22 parametrů, dimenzí bude také 22. Jako počáteční nastavení jsou upřednostněny všechny parametry.

Počet jedinců se odvíjí od počtu dimenzí. Obvykle se volí podle vzorce 2.2.

Komunikace se nejjednodušeji provádí předem zvoleným počtem jedinců, kteří spolu komunikují. Na počátku jsou zvoleni 4 jedinci pro jedno okolí (vzájemná komunikace).

Jedinci, kteří v průběhu opustí prostor možných řešení, jsou náhodně umístěni v prostoru. Tento jev je kontrolován za pomoci rozsahu minimální a maximální rychlosti ve všech dimenzích. Jednotlivé rozsahy se shodují s rozsahy parametrů simulátoru TORSC.

4.4.2 Volba koeficientů

Koeficient w , který řídí rychlost, je zvolen na začátku algoritmu na hodnotu 0,9. Postupem času se jeho hodnota snižuje až na hodnotu 0,4. Snižování koeficientu rychlosti se děje z důvodu nutné pomalejší konvergence ke konci algoritmu.

Při volbě náhodného generování koeficientů φ_1 , φ_2 není zapotřebí hodnoty volit. Při možnosti volby koeficientů c_1 a c_2 , by bylo vhodné zvolit jako počáteční nastavení větší hodnotu u c_1 . Postupem času by se měly hodnoty srovnat a c_2 bude ke konci větší. Algoritmus by měl ze začátku hledat lokální extrémy a ke konci konvergovat do globálního extrému. Je velice pravděpodobné, že množina možných řešení obsahuje více možných řešení, poté je možné více upřednostňovat koeficient c_1 .

4.5 Hybridní algoritmus

U optimalizační techniky spojené z genetického algoritmu a optimalizace hejnem částic není zapotřebí navrhovat žádné nové parametry. Odlišnost se týká populace přesněji řečeno ve výběru jedinců, jelikož do algoritmu vstupuje vždy lépe ohodnocená polovina populace. Druhou odlišností případně rozšíření je nutnost seřazení vždy nově vzniklé populace.

4.6 Ukládání výsledků

V rámci dalšího zpracování je nutné vhodně ukládat výsledky⁵ optimalizačních technik. Nejvhodnější formou bude ukládání veškerých dostupných informací. Tedy ukládat veškeré parametry vozu a k nim vrácené informace o nejlepším času na kolo, maximální rychlosti, ujeté vzdálenosti a poškození vozu.

Vhodný formát souborů pro uložení se nabízí csv. Veškeré soubory jsou vždy pojmenovány *aktuální-datum_zvolená-optimalizační-technika_charakteristika-trati_parametr.csv*.

Aktuální datum je zřetelně daná položka podobně jako zvolená optimalizační technika. Charakteristikou trati se rozumí převážně povrch trati (asfalt, šotolina, sníh, led atd.), pod položkou parametr se skrývá nastavení určitého parametru optimalizačního algoritmu, které je například netypicky zvolené.

⁵ Přesněji by se dalo říci spíše mezivýsledky, výsledky práce jsou nejlepší nastavení.

4.7 Shrnutí

Kapitola představila návrh řešení problému nastavení parametrů závodního vozu za pomoci zvolených optimalizačních technik. Kapitola obsahuje řadu nejistot, které jsou zakončeny zvolením počátečního nastavení. Je to z důvodu, které vnáší samotné optimalizační techniky, které jsou inspirovány přírodou. Jistotu konečného nastavení parametrů dává až implementace.

Může se zdát divné, proč se s každým parametrem pracuje v jeho skutečném rozsahu a poté se převádí (mapuje) do rozsahu intervalu $\langle 0, 1 \rangle$. I s ohledem na generování náhodných (pseudonáhodných) čísel se bude lépe pracovat se skutečnými rozsahy.

5 Implementace

V následující kapitole budou popsány důležité části algoritmů jednotlivých optimalizačních technik, které jsou představeny v kapitole 2. Na úvod bude naznačena komunikace mezi částí obsahující optimalizační algoritmus (klient) a částí přijímající jednotlivá nastavení (server). V rámci realizace projektu byl vybrán implementační jazyk Java. Také je nutné použít program TORCS. V projektu je použita verze 1.3.1.

Seznam souborů se zdrojovými kódy:

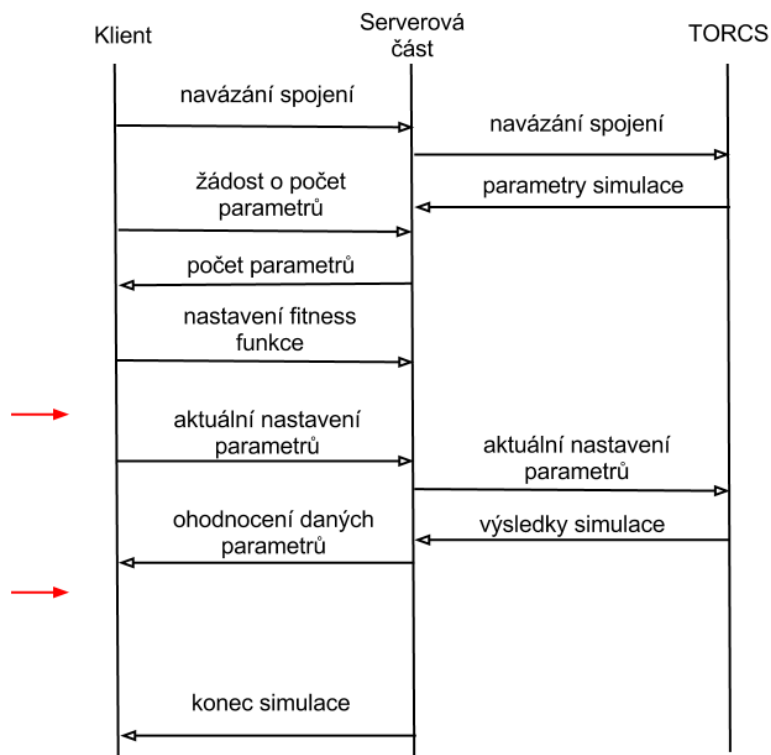
- evaluator
 - ServerCommunication.java
 - FitnessFunction.java
 - TimeOverException.java
- „zvolená technika“ (ga, pso, hgapso)
 - ga.java, pso.java, hgapso.java
 - fitness.java

Veškeré zdrojové kódy jsou převzaty ze šablony projektu CIG [21]. U genetického algoritmu je kód upraven z původní verze do potřeb práce. Ostatní techniky (pso, hgapso) jsou sepsány autorem této práce. Vše je uvedeno v hlavičkách všech souborů se zdrojovými kódy.

5.1 Komunikace

Nejdříve bude popsána komunikace mezi serverovou a klientskou částí. Serverovou částí je myšlen program komunikující s programem TORSC od něhož získává jednotlivé výsledky a přeposílá zpět klientské části. Celková komunikace je naznačena na obrázku 5.1. Veškeré úkony, které se nacházejí mezi červenými šipkami se neustále opakují, dokud nevyprší čas simulace.

Nejdříve je nutné poslat požadavek na server pro zahájení komunikace, kde se zašle ip adresa spolu s číslem portu. Server naváže komunikaci s klientem a rovněž s programem TORCS. V rámci komunikace klient server se zasílá informace o aktuálních parametrech vozu a zbývajícím časem simulace. Server zasílá programu TORCS nastavení vozu, nazpět dostane 4 parametry a sice nejlepší čas na kolo, ujetou vzdálenost, maximální rychlost a poškození vozu. Klientovi se vrací informace o fitness hodnotě aktuálního nastavení. Čas simulace je hlídán na straně serveru, přičemž klient si jej může vyžádat metodou *getRemainingTime* a na základě návratové hodnoty může ukončit optimalizační algoritmus. Poté veškerá komunikace skončí.



Obrázek 5.1 Komunikace

5.2 Fitness funkce

Fitness funkce spočívá v nastavení váhových koeficientů u jednotlivých parametrů (nejlepší čas na kolo, maximální dosažená rychlost, ujetá vzdálenost poškození), které vrátí program TORCS. Zde se nedá přesně určit, jakou váhu bude mít který parametr, ale zcela jistě bude mít největší váhu parametr nejlepšího času na kolo. Další 3 parametry jsou víceméně na stejné úrovni.

Velice důležité je zvolit, které fitness funkce se bude používat. Volba probíhá v hlavní metodě na straně klienta, přičemž nastavení je aplikováno na straně serveru pomocí metody *setFitnessFunction*.

5.3 Optimalizační algoritmus

Podkapitola optimalizační algoritmus představuje skladbu všech použitých algoritmů. Veškeré informace jsou zde v obecné rovině a pro lepší představu jednoho cyklu algoritmu. Jednotlivé cykly jsou podrobně popsány v následujících kapitolách (5.4 Genetický algoritmus, 5.5 Optimalizace hejnem částic, 5.6 Hybridní algoritmus). Nyní již obecný pseudokód optimalizačního algoritmu:

```

    inicializace_spojení_a_parametrů_simulace;
    for(i = 0; i < počet_generací OR vypršel_čas_simulace; i++)
    {
        ohodnocení_jedinců;
        průběh_daného_algoritmu;
    }

```

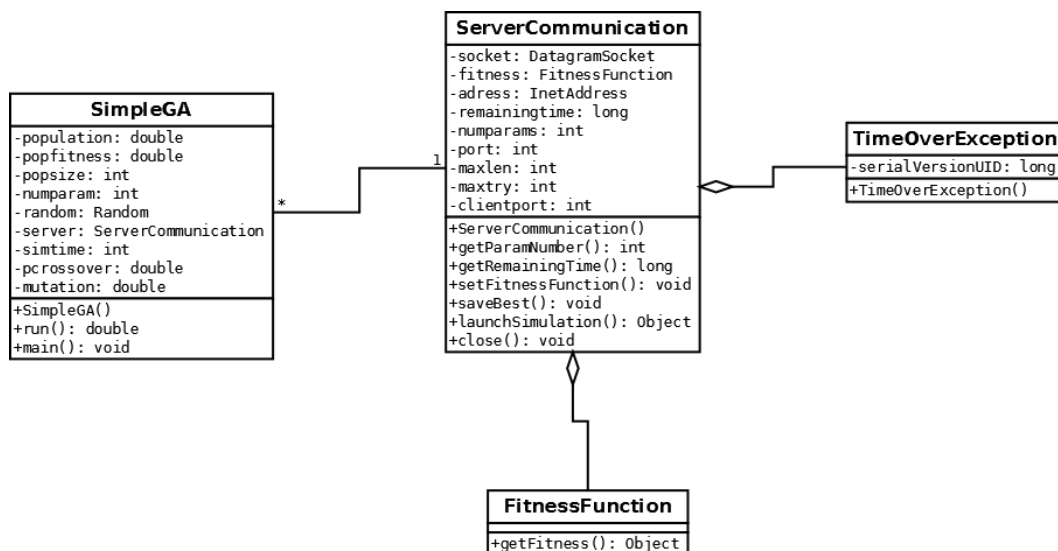
5.4 Genetický algoritmus

Genetický algoritmus (soubor SimpleGA.java) je reprezentován třídou *SimpleGA*. Třída obsahuje dvourozměrné pole *population*, kde je nutné uložit jednotlivé jedince (každý má svoje parametry nastavení závodního vozu). Za zmínku zcela jistě stojí pole *popfitness* obsahující ohodnocení jednotlivých jedinců s aktuálním nastavením.

Při vytvoření objektu třídy *SimpleGA* se alokují zmíněná pole, především se naplní populace náhodně vygenerovanými hodnotami, které jsou omezeny rozsahem hodnot jednotlivých parametrů závodního vozu.

Hlavní metoda třídy *SimpleGA* se nazývá *run*, má jeden parametr a sice počet generací genetického algoritmu. Pokud vyprší čas simulace, algoritmus se ukončí dříve. Jako první (v rámci cyklu) přichází na řadu ohodnocení jedinců, kdy se na server pošlou údaje aktuálního jedince spolu s časem simulace. Vráti se hodnota přímo z fitness funkce, která se přímo uloží na index odpovídající indexu daného jedince v populaci. Další důležitou částí je selekce jedinců, která je realizována turnajem. Náhodně vybraní jedinci jsou uloženy do pole *selected* za pomoci porovnání jejich fitness hodnot. Vždy se vyberou dva jedinci a uložen je pouze jedinec s lepší fitness hodnotou. Na tyto jedince je aplikován operátor křížení. Například křížení uniformní kdy se na základě náhodné hodnoty vybírá parametr z prvního či druhého jedince. Výsledek křížení je v poli *recombined*. V rámci pole *recombined* je aplikován i operátor mutace, kdy je vygenerováno číslo, na jehož základě je anebo není aplikován operátor na aktuálního jedince. Jako poslední v cyklu (viz kapitola 5.3) je zapotřebí vybrat jedince do nové populace. Zde se opět nabízí turnajový výběr.

Během algoritmu se ukládají veškeré mezivýsledky a na konci algoritmu se zobrazí (uloží) nejlepší nalezené řešení.



Obrázek 5.1 Diagram tříd - Genetický algoritmus

5.5 Optimalizace hejnem částic

Algoritmus je zastoupený třídou *clasicPSO* (soubor *clasicPSO.java*). Podobně jako u genetického algoritmu je populace (zde hejno) realizována dvourozměrným polem, které je nazváno *swarm*. Také je nutné mít pole pro udržování fitness hodnot jedinců hejna (*swarmfitness*). Pole *pBest* a *gBest* slouží k udržení aktuálně nejlepší hodnoty fitness daného jedince potažmo skupiny jedinců, kteří spolu komunikují. Pro stejný účel slouží pole *pBestLoc* a *gBestLoc* s rozdílem ukládání nejlepších hodnot nastavení závodního vozu respektive pozice jedince v prostoru. Poslední důležité pole je *communicate*, které slouží k uchování spolu komunikujících jedinců hejna tj. jejich indexů. Za zmínku stojí konstanta *PART_COM* značící právě počet jedinců ve skupině. S tímto souvisí i alokace polí *gBest* a *gBestLoc*. Zatímco u *pBest* a *pBestLoc* je nutné pole alokovat na velikost hejna u zmíněných polí („g“) stačí alokovat na velikost hejna dělené konstantou *PART_COM*.

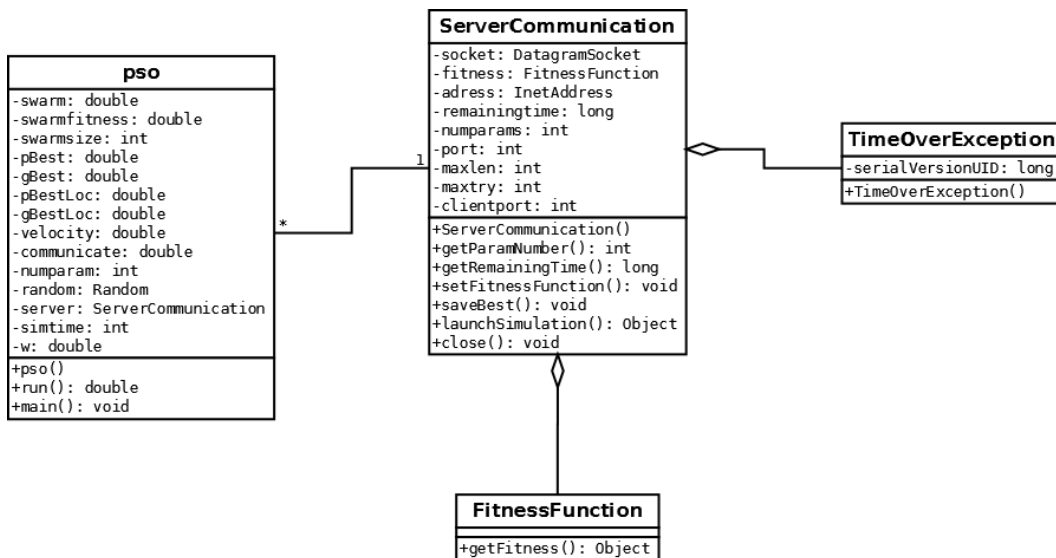
Veškeré alokace jsou provedeny v konstruktoru při vytvoření objektu třídy *clasicPSO*. Zde také probíhá vygenerování náhodných hodnot, omezených spodní a horní hranicí daného parametru závodního vozu, jedinců. Do polí označující dosavadní nejlepší fitness se uloží hodnota *Double.MIN_VALUE* (nejmenší hodnota datového typu *double*). Je to z důvodu správné aktualizace hodnot (fitness, pozice jedince) již od začátku algoritmu. V konstruktoru dochází i k naplnění pole *communicate* za pomoci vektoru. Vektor se na začátku naplní čísly od 0 do čísla odpovídajícímu velikosti hejna. Poté v cyklu se vygeneruje náhodné číslo od 0 do aktuální velikosti vektoru, které je poté uloženo na další pozici v poli *communicate*. Tento proces se opakuje dokud není vektor prázdný:

```

while(je_vektor_neprázdný)
{
    communicate[i] = vektor[náhodné_číslo(0, aktuální_velikost_vektoru-1)];
    odstranění_přiřazeného_prvku_z_vektoru;
    i++;
}

```

Samotný algoritmus optimalizace hejnem částic je obsažen v metodě *run*. Algoritmus je ukončen rovněž počtem iterací, který je zároveň parametrem metody, anebo dříve pokud vyprší čas simulace. Na začátku každého cyklu (viz kapitola 5.3) je nutné jedince ohodnotit. Na server se vždy posílá aktuální jedinec hejna spolu s časem simulace. Po provedení se uloží návratová hodnota, a sice fitness daného jedince (do pole *swarmfitness*). V případě lepšího řešení než je aktuální uložené v *pBest*, následuje aktualizace hodnot (fitness, pozice) v rámci daného jedince. Ihned poté nastává aktualizace nejlepší nalezené hodnoty v rámci daných skupin jedinců. Zde se porovnává vždy nejlepší hodnota fitness jedince ve skupině s aktuální nejlepší hodnotou skupiny. Po aktualizacích proběhne výpočet nových hodnot parametrů podle vzorců 2.3 a 2.4.



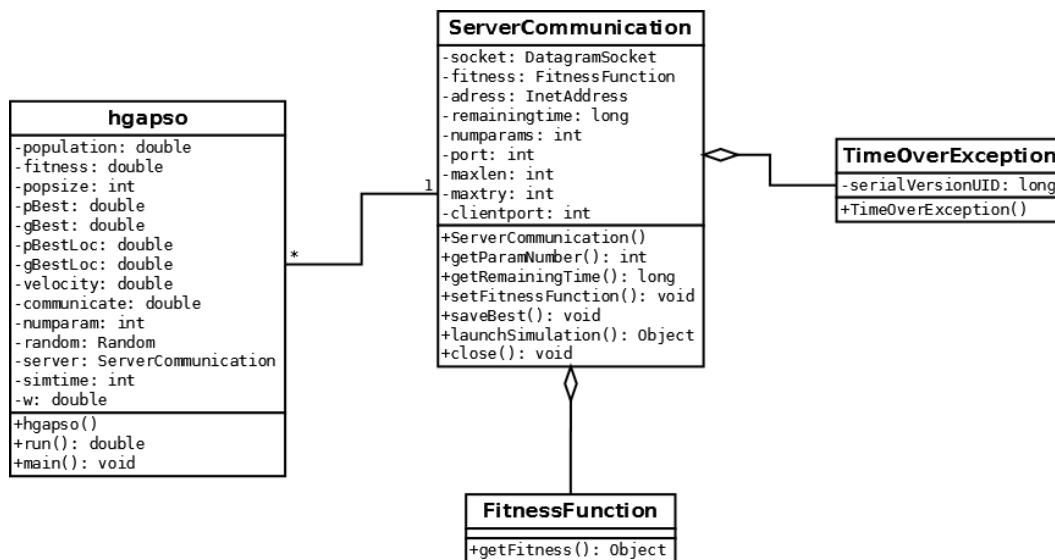
Obrázek 5.2 Diagram tříd - Optimalizace hejnem částic

5.6 Hybridní algoritmus

Spojení genetického algoritmu a optimalizace hejnem částic (soubor *hgapso.java*) reprezentuje třída *hgapso*. V rámci zdrojového kódu jsou použity stejné proměnné, které jsou popsány v předchozích dvou kapitolách. Důležitou změnou je využití populace (reprezentované polem *population*), kdy v každém cyklu algoritmu (viz kapitola 5.2) je použita pouze polovina zmíněného pole. Jak již bylo

zmíněno v kapitole 2.4, po ohodnocení jedinců je důležité populaci seřadit podle fitness hodnot jedinců a právě lepší polovina vstupuje do další iterace algoritmu. Po dokončení algoritmu PSO se aktualizují pozice jedinců a uloží se do první poloviny pole *population* a právě tato polovina je použita jako vstup genetického algoritmu. Výstup genetického algoritmu je uložen do druhé poloviny pole *population*.

Populace je seřazena za pomoci typu *SortedMap* a následně pomocí *Iteratoru* získána lepší polovina populace.



Obrázek 5.3 Diagram tříd - Hybridní algoritmus

5.7 Shrnutí

Kapitola popsala důležité aspekty zdrojových kódů jednotlivých optimalizačních technik, které jsou použity v rámci hledání nastavení vozu pro určité okruhy. Všechny techniky byly otestovány nejdříve samostatně bez komunikace se serverem, kde na místo skutečných fitness hodnot byly použity předem definované. Popsaný postup byl zvolen především k odladění chyb. Poté byly algoritmy otestovány v plném rozsahu.

6 Dosažené výsledky

Kapitola popisuje jednotlivé výsledky dosažené za pomoci optimalizačních metod, které byly zvoleny v návrhu práce. Struktura kapitoly je rozdělena v rámci zvolených tratí především dle jejich povrchů, který hraje klíčovou roli pro nastavení.

Zvolená trať je vždy popsána v rámci svojí charakteristiky a rámcového (předpokládaného) nastavení vozu. Následně jsou uvedeny výsledky v podobě nejlepšího nalezeného nastavení zvolených technik. Veškeré zde zvolené okruhy jsou obsaženy v programu TORCS. Výsledky v podobě časů jsou vždy uvedeny u daného okruhu. Vždy je uveden čas pro výsledek genetického algoritmu (GA), optimalizace hejnem částic (PSO), hybridního algoritmu (HGAPSO) a času, který odpovídá zjetí kola bez chyby (MANUAL). Přesné výsledky jednotlivých parametrů jsou k nalezení v příloze práce (příloha 2 – Nalezená nastavení).

6.1 Obecné zhodnocení

Nejdříve je zapotřebí zmínit pár poznatků, které budou společné pro veškeré okruhy. Především se jedná o nastavení optimalizačních technik a určitých parametrů programu TORCS.

Nejdůležitější zmínkou je nastavení (na straně klienta) určující dobu vyhodnocení jednotlivých řešení v aktuální generaci. Prakticky se jedná o rozdělení celkového přiděleného času serveru. Je nutné nalézt kompromis, protože pokud se zvolí doba moc malá, tak optimalizační technika proběhne ve více generacích, ale jednotlivá nastavení nebudou dostatečně otestována (ohodnocena), protože se nestačí zjet potřebný počet kol nebo vůbec žádné. Tudíž algoritmus se může ubírat špatným směrem. Na druhou stranu pokud se zvolí doba ohodnocení moc velká, tak sice otestování (ohodnocení) jednotlivých nastavení proběhne důkladně, ale optimalizační technika proběhne v malém počtu generací. V průběhu testování nastaly situace, kdy algoritmus proběhl pouze ve 2 generacích. Od rozdělení simulačního času se také odvíjí parametr počet jedinců v populaci (případně hejnu), jelikož pokud bude počet moc velký, poté bude zbytečně vyčerpán čas simulace. Z těchto poznatků plyne důležitý závěr, spíše než nastavení jednotlivých optimalizačních technik, je důležité správně vyladit čas simulace pro jedno ohodnocení na serveru. Po celkovém testování níže zvolených tratí by se dalo v procentech vyjádřit, že správné nastavení simulace je důležité z 80%. Zbylých 20% tvoří nastavení optimalizační techniky. Velkým problémem jsou delší okruhy, kde se časy na kolo pohybují nad jednu minutu. Důvodem je větší spotřeba času simulace na jedno ohodnocení a již se nedostává k aplikaci operátorů optimalizačních algoritmů ve více generacích. Byly otestovány možnosti od 100 do 20 000. Ideálně se jevíly hodnoty kolem 7 000 na jedno ohodnocení. Nastavení optimalizačních technik bylo zvoleno podle běžných parametrů, jak je uvedeno v kapitole 2 Optimalizační techniky.

Různé experimenty s nastavením algoritmů nevedly k lepším výsledkům. Toto zjištění je přisouzeno k malému počtu iterací optimalizačních algoritmů. Při hodnotách (parametru simulace) od 6 do 8 tisíc je počet generací cca 4, což je velmi malé číslo. Avšak i původní projekt nepočítal s vysokým počtem generací (byl nastaven na 20 a extrémně malou dobu simulace 600).

Dalším problémem byla samotná fitness funkce, jelikož musel být vyloučen parametr určující čas kola, který ve většině případů obsahuje hodnotu -1. S velkou pravděpodobností se jedná o problém zmíněný v kapitole 3.4.4 (navrácení hodnoty -1 při nedokončení prvního kola), jelikož při zvýšení času simulace (jedno ohodnocení) například na hodnotu 10 000 se začnou ve větší míře vracet i časy kol. Ovšem při tak vysoké hodnotě se prakticky neprojeví vliv optimalizačního algoritmu, protože proběhne v maximálně 2 generacích. Tudíž hodnota času kola není vhodným parametrem pro určení fitness hodnoty. Avšak problém nemusí být pouze v parametru simulace, ale i v komunikaci mezi programem TORCS a serverovou částí, jelikož výpisy časů kol jsou v programu TORCS v pořádku. Tato nepříjemnost, byla odstraněna preferováním nejvyšší dosažené rychlosti, jelikož v rámci pozorování výsledků (výpis v TORCS) se hodnoty shodovaly. Při zlepšení času na kolo, byla výrazně vyšší maximální rychlost. Také byla odzkoušena varianta fitness hodnoty maximální rychlost mínus poškození vozu, aby se předešlo situacím vysoké rychlosti, která měla za následek poškození vozu. Tento fakt měl zanedbatelné výsledky oproti hodnotě fitness tvořené pouze z maximální rychlosti vozu.

Parametr	Díl	Název	Hodnota
1	Převodovka – převod 2	poměr	3.5
2	Převodovka – převod 3	poměr	4.0
3	Převodovka – převod 4	poměr	4.5
4	Převodovka – převod 5	poměr	3.75
5	Převodovka – převod 6	poměr	3.5
6	Zadní křídlo	úhel	5
7	Přední křídlo	úhel	3
8	Brzdy	poměr přední-zadní	0.55
9	Brzdy	maximální tlak	50000
10	Přední stabilizátor	pružina	4500
11	Zadní stabilizátor	pružina	4000
12	Přední kola	světla výška	100
13	Přední kola	sblhavost	0.5
14	Přední kola	odklon	-3.5
15	Zadní kola	světla výška	120
16	Zadní kola	odklon	-2.0
17	Přední zavěšení	pružina	9000
18	Přední zavěšení	výška	0.1
19	Zadní zavěšení	pružina	8000
20	Zadní zavěšení	výška	0.05
21	Přední brzdy	průměr kotouče	340
22	Zadní brzdy	průměr kotouče	250

Tabulka 6.1 Nastavení zvolené člověkem

Důležitým faktorem je také důvěryhodnost samotného simulátoru TORCS. V rámci práce bylo uskutečněno dotázání respondentů, kteří mají znalosti z oblasti strojírenství. Na jejich nastavení vrátil simulátor velice špatné výsledky. V průměru od 30 sekund do 2 minut rozdílu oproti časům, které

byly bezchybně odjety v simulátoru (myšleno doopravdy odjety pomocí uživatele). Zde je ukázka nastavení zvoleného člověkem a výsledné časy. Nastavení (tabulka 6.1) je ukázkou pro rychlý asfaltový okruh, který je zároveň náročný na brzdy, čili kde se brzdí z vysokých rychlostí do velmi malých během krátké doby. Tyto fakty splňuje okruh Forca (představen v kapitole 6.2.1). Ovšem po zaslání hodnot do simulátoru byly navraceny časy cca o minutu horší. Podobně na tom byly i ostatní okruhy, proto nejsou jednotlivá nastavení v práci dále uvažována a je zapotřebí jednotlivé výsledky brát s rezervou. Důvod je prostý, i kdyby zmíněné nastavení nebylo neoptimálnější, skutečný rozdíl by byl v řádu sekund nikoliv desítek sekund či dokonce minut. Ovšem v rámci objektivitu je nutné přiznat, že zvolené nastavení je potřeba brát s jistou dávkou subjektivitu autora práce (platí i pro další objasnění nastavení u zvolených tratí) a respondentů.

Avšak získané výsledky je nutné porovnat s určitými daty, jinak práce ztrácí na významu. Proto byl vyhrazen čas pro zjetí všech zvolených časů přímo v simulátoru. Výsledné časy představují dokonale zjetý okruh⁶ bez chyby jsou zapsány do sloupce MANUAL v každé kapitole představující okruh a získané výsledky.

6.2 Asfaltové okruhy

Okruhy popsané v kapitole 6.2 se nacházejí v programu TORCS pod „Road trucks.“

6.2.1 Forca

Okruh (obrázek 6.1) je velice podobný skutečnému okruhu v italské Monze. Start se nachází na velmi dlouhém rovinatém úseku (šipka označuje umístění a směr závodění). Oproti skutečné trati chybí kombinace pomalých zatáček, které jdou ihned po sobě tzv. „šikana.“ Následující zatáčka (první na trati Forca) je na pravou ruku a projíždí se na plný plyn. Po této zatáčce chybí opět kombinace pomalých zatáček. Tento fakt by se měl projevit především v nastavení brzd, které nebudou tolik namáhané a nebude potřeba nastavovat poměr brzd (výrazněji) na přední část vozu. Poté následuje dvojice ostřejších levých zatáček, kde je namáhána levá část vozu, především na levé přední pneumatice dochází k většímu opotřebení. Po delším rovinatém úseku se nachází kombinace levé, pravé a levé zatáčky, která klade velké nároky na stabilitu. Je důležité neztratit rychlost, protože z popsané kombinace zatáček se najíždí na delší rovinatou část. Nastavení podvozku v této části bude hrát klíčovou roli, například tlumiče budou zcela jistě tvrdší a světlá výška vozu co nejnižší. Poslední zatáčkou je tzv. „parabolica,“ kde na jezdce působí delší dobu přetížení okolo 4G⁷. Poté následuje cílový rovinatý úsek.

6 Nikdy se nejedná o první kolo, jelikož tento čas je logicky pomalejší z důvodu startu z místa.

7 G jsou násobky váhy v tomto případě samotného jezdce, které na něj působí.



Obrázek 6.1 Okruh Forca [33]

Výsledky pro okruh Forca jsou uvedeny v tabulce 6.2. U okruhu Forca se nepodařilo přiblížit optimálnímu času, který je uveden ve sloupci MANUAL. Pro tuto trať platí uvedené fakta v kapitole 6.1 Obecné zhodnocení, především fakt s nastavením parametru délky ohodnocení na serveru.

Co se týká jednotlivých parametrů, tak lze vidět velké nesrovnalosti v oblasti převodovky, kdy jednotlivé převodové stupně mají větší rozdíly a také jsou v určité disharmonii. Dále je možno zmínit nastavení brzd. Nastavení brzdového tlaku se zdá být relativně v pořádku, ovšem rozložení brzd je nastaveno výrazně na zadní část, což může mít za následek přetáčivost vozu. V případě nalezeného nastavení PSO je rozložení mírně na přední část, což je ideální, jelikož setrvačná síla směřuje (při brzdění) na přední část vozu. Pouze u genetického algoritmu je v pořádku nalezené nastavení průměrů kotoučů brzd, jelikož na předních brzdách jsou výrazně větší. Zcela jistě stojí za zmínku nastavení světlé výšky na přední a zadní části vozu, kde jsou velké rozdíly.

GA	PSO	HGAPSO	MANUAL
01:40:36	01:43:23	01:55:19	01:31:31

Tabulka 6.2 Časy na kolo pro okruh Forca

Celkově nalezené nastavení vozu není dobré, což je způsobeno menším počtem generací jednotlivých algoritmů. Podle tabulky nejlépe dopadl genetický algoritmus, ale rozdíl je 9 sekund oproti ideálně zajetému času, což je rozdíl třídy.

6.2.2 CG Speedway number 1

Okruh (obrázek 6.2) má jednu zvláštnost a sice start (šipka označuje umístění a směr závodění) ze zatáčky, což nebývá pravidlem. Druhou zvláštností jsou asfaltové únikové zóny. Ovšem tyto zóny tvoří velmi hrbolatý povrch, tudíž se nevyplatí vyjet mimo vytyčené území trati. Mohlo by dojít k poškození vozu. První část trati tvoří rovinný úsek, který je následován zatáčkou na levou ruku. Do této zatáčky je nutné mírně přibrzdit, ihned při výjezdu začíná táhlá pravá zatáčka, která se jede na plný plyn. Poté následuje nejobtížnější pasáž okruhu a sice dvě po sobě rychle jdoucí levé zatáčky, přičemž druhá se neustále zavírá. Zde je nutný dobrý výjezd jelikož poté následuje rovinný úsek, kde je potřeba nabrat rychlost. Poslední je zatáčka na levou ruku, ve které se nachází start (cíl) okruhu.

Celkově není trať náročná na nastavení vozu. Rovinné úseky nepatří k nejdelším a jednotlivé zatáčky k nejobtížnějším, tudíž nebude nutné řešit problémy s přitlakem.



Obrázek 6.2 Okruh CG
Speedway 1 [33]

Okruh CG Speedway 1 patří ke kratším a jsou vidět menší rozdíly v časech na jedno kolo (tabulka 6.3), což potvrzuje obsah kapitoly 6.1, konkrétně problém s nastavením parametru ohodnocení.

V parametrech genetického algoritmu jsou vidět stále větší rozdíly v nastavení převodových stupňů, ale oproti předchozí trati jsou tyto rozdíly přijatelné. U ostatních dvou technik je převodovka relativně dobře nastavená. Ovšem pružiny jsou nastavené jako velmi měkké u všech tří technik. Nastavení by se hodilo spíše na štěrkový povrch. Podobně je nastavená i světlá výška, která se pohybuje v horní části intervalu a to znamená nastavení opět na štěrk.

GA	PSO	HGAPSO	MANUAL
42:78	44:16	41:91	40:78

Tabulka 6.3 Časy na kolo pro okruh CG1

Celkově je ale nastavení v principu podobně nevhodné, jako u předchozího okruhu, ovšem výsledek je daleko lepší. Tento fakt nahrává mírné nedůvěryhodnosti simulátoru TORCS.

6.2.3 CG track 2

Oproti předchozí je aktuální trať (obrázek 6.3) náročná, jak svým profilem, tak i volbou správného nastavení vozu. Únikové zóny tvoří tráva a štěrk. Úvodní pravotočivé zatáčky předcházejí složitější kombinaci levých zatáček. Zde již je zapotřebí větší přitlačná síla, čili světlá výška bude nižší a zadní křídlo pod větším úhlem. Ovšem úhel není možno nastavit moc velký, jelikož na výjezdu je rovinný úsek, kde je zapotřebí nabrat rychlost. Na jeho konci se brzdí do nejtěžší pasáže okruhu, který se skládá z ostřejší pravotočivé a táhlejší levotočivé zatáčky. I zde je zapotřebí více přitlaku. Následuje úsek, který končí poslední zatáčkou okruhu, zde je důležité dobré najetí, jelikož je potřeba nabrat rychlost do dalšího kola (cíle).



Obrázek 6.3 CG track 2 [33]

První okruh u kterého určitá optimalizační technika našla lepší nastavení vozu, které způsobilo lepší čas než ten, který byl dosažen po zajetí bezchybného kola.

Problém převodovky zde není tak velký. Také je zapotřebí zdůraznit správný poměr průměrů kotoučů brzd u všech technik. U hybridního algoritmu (HGAPSO) je velmi dobrý poměr rozložení brzd, kde je vidět nastavení spíše na přední brzdy. Naopak u genetického algoritmu je nastavení na brzdy zadní. Tento fakt zcela jistě způsobí přetáčivost vozu v každé zatáčce. Co je také nepříjemné u všech technik je nastavení světlé výšky, kdy je vůz v přední části téměř 2x vyšší než v části zadní.

GA	PSO	HGAPSO	MANUAL
1:00:23	55:84	57:96	56:45

Tabulka 6.4 Časy na kolo pro okruh CG2

Takto nastavený vůz bude velmi nestabilní a k nápravě nepomůže slušné nastavení přítlačných křidel u hybridní techniky (HGAPSO). Velmi slušně jsou nastavené pružiny a tlumiče, které jsou zvoleny jako tvrdší, což je zapotřebí k dosažení lepší stability na asfaltovém povrchu.

6.2.4 Olethros Road 1

Trať (obrázek 6.4) patří k nejnáročnějším, což zákonitě platí i pro nastavení jednotlivých parametrů závodního vozu. Okruh jako takový je plný nerovností, ale má i rychlé pasáže. Nastavení podvozku z hlediska stability versus rychlosti vozu bude plně kompromisů. První zatáčka je náročná na brzdový systém a je nutná velice dobrá stabilita vzhledem k rychlosti dosažené na rovinném úseku a poslední zatáčky, která se projíždí na plný plyn. Následuje klopená zatáčka na levou ruku, kde je velice namáhána pravá část vozu po delší dobu. Ovšem ihned na výjezdu jsou výrazné nerovnosti přímo na trati. Tyto fakty ovlivňují nastavení tlumičů a pružin. V okolí první zatáčky by přišlo vhod nastavení tužších tlumičů a pružin, naopak na nerovnostech by při takovém nastavení došlo k poškození vozu. Zde jsou zapotřebí (tlumiče a pružiny) měkčí a také vyšší, což zvýší světlou výšku vozu. Od výjezdu z klopené zatáčky až po nájezd do poslední, je trať pokrytá nerovnostmi a náročnými zatáčkami, čili

nastavení bude směřovat spíše k měkčím tlumičům, ovšem nepůjde do extrému (závodí se pořád na asfaltu nikoliv na šotolině), jelikož by se vůz mohl stát přetáčivým.



Obrázek 6.4 Olethros Road 1 [33]

U aktuální trati se opět projevuje problém nastavení ohodnocení na serveru. Okruh Olethros je navíc velmi náročný a tyto kombinace způsobí velmi špatné časy (tabulka 6.5).

Na začátek opět zmínka k nastavení převodovky, které se zdá být velmi slušné u hybridní techniky naopak velmi špatné (velké rozdíly v délce převodů) u genetického algoritmu. Velice solidní je nastavení pružin, které jsou cca ve středu intervalu. Trať je sice asfaltová, ale obsahuje velkou řadu nerovností. Podobně dobré je nastavení světlé výšky vozu. Velký nedostatek je v přitlačných křídlech, které jsou nastaveny na malý úhel. Toto nastavení způsobí malý přitlak celého vozu, ovšem na podobně náročné trati je zapotřebí přitlak větší (lepší stabilita vozu).

Obecně je nastavení vozu srovnatelné s prvním okruhem a i výsledky časů jsou velmi špatné.

GA	PSO	HGAPSO	MANUAL
2:25:52	2:18:04	2:21:94	1:56:58

Tabulka 6.5 Časy na kolo pro okruh Olethros

6.2.5 Street 1

Okruh (obrázek 6.5) je typickou městskou tratí. Podobné okruhy jsou velice obtížné pro jezdce, jelikož neexistují (až na výjimky) únikové zóny. Trať lemují pouze svodidla. Jednotlivé zatáčky mají poměrně velké obrubníky. Nevyplatí se obrubníky přejíždět, může dojít k poškození vozu, případně se vůz dostane do smyku. Městské okruhy se vyznačují pomalými zatáčkami a je nutné nastavit kratší poměry převodových stupňů z důvodu lepší akcelerace při výjezdu ze zatáček. Okruh Street 1 není výjimkou, jak je vidět na obrázku 6.5. Okruh je pokrytý relativně rovným asfaltem, čili světlá výška vozu se může nastavit co nejnižší. Nižší světlá výška vozu bude mít za následek lepší přitlak, který je nutný v kombinaci zatáček ihned po startu. Jedná se o rychlou kombinaci 3 zatáček. Následující zatáčky vyžadují již zmíněnou akceleraci. Okruh má také rovinný úsek s jednou levotočivou zatáčkou, která se ovšem projíždí na plný plyn. Na konci (rovinnatého úseku) se brzdí do velmi pomalé

zatáčky tzv. „vlásenky“⁸.“ Poslední zatáčka okruhu má velmi podobné vlastnosti a opět je nutná dobrá akcelerace z důvodu navýšení co nejlepší rychlosti do dalšího okruhu.



Obrázek 6.5 Street 1 [33]

Okruh patří k těm delším, čili se opět projeví problém v nalezení kompromisu mezi počtem generací optimalizačního algoritmu a parametrem ohodnocení. Podle výsledků časů je vidět opětovný rozdíl mezi bezchybně zajetým kolem a nalezeným řešením optimalizačních technik.

Ohledně parametrů závodního vozu je vše podobné jako u předchozích tratí. Ovšem více okruhů potvrdí tvrzení problému ohodnocení (viz kapitola 6.1).

GA	PSO	HGAPSO	MANUAL
1:30:72	1:34:74	1:29:74	1:20:57

Tabulka 6.6 Časy na kolo pro okruh Street 1

6.2.6 Wheel 2

Trat' Wheel 2 (obrázek 6.6) je velice podobný závodnímu okruhu v japonské Suzuce. Trat' má určitá specifika, jedním z nich je poměrně úzká trat' nebo velké únikové zóny, které jsou ovšem šterkové. Vyjetí z dráhy může znamenat konec závodu. Velkou zvláštností je překřížení trati, které je vyřešeno přemostěním. Co se týče nastavení vozu, trat' vyžaduje velmi dobrý přítlak, jelikož je složena z kombinace středně rychlých zatáček. Obecně je celá trat' velice rychlá, ale nabízí i úskalí v podobě „vlásenky“ nacházející se za křížením a také „šikanu“ (předposlední zatáčka před cílem).



Obrázek 6.6 Wheel 2 [33]

⁸ Pomalá zatáčka, která se otáčí prakticky o 180 stupňů.

Okruh je spolu s tratí Olethros nejnáročnější a také nejdelší s testovacích tratí. Nemá cenu zde opakovat pořad dokola stejné fakty. Výsledky v podstatě potvrzují kapitolu 6.1.

GA	PSO	HGAPSO	MANUAL
2:15:31	2:28:37	2:21:63	1:56:29

Tabulka 6.7 Časy na kolo pro okruh Wheel 2

6.2.7 Alpine 2

Okruh vedoucí v horách. Obsahuje typické aspekty podobného místa jako tunel, most, silnice vedená po kraji hor apod. Závodí se na suché silnici, ovšem okruh má nulové únikové zóny. Také jsou zatáčky nepříjemné v podobě neviditelnosti výjezdů. Tento fakt ovlivňuje správné najetí respektive celkové projetí zatáčkou, jelikož je obtížné dostat se správně na střed zatáčky. Obtížnosti celkové trati také nahrává její profil. Trať neustále stoupá a klesá, tudíž aspekty jako nedotáčivost respektive přetáčivost vozu zde budou hrát větší roli. Okruh je tvořen kombinacemi rovinatých úseků a pomalých zatáček, čili je velice náročný na brzdy. Například pasáž ve druhé části trati, kde je zatáčka přesně pod úhlem 90 stupňů je velice obtížná, jelikož se zde najíždí do tunelu. Na podobných místech reálně hrozí výrazné poškození vozu tzn. odstoupení ze závodu.



Obrázek 6.7 Alpine 2 [33]

Výsledky jsou podobné jako u předchozí trati. Okruh je uvedený pouze pro potvrzení kapitoly 6.1.

GA	PSO	HGAPSO	MANUAL
1:43:93	1:45:67	2:07:18	1:35:44

Tabulka 6.8 Časy na kolo pro okruh Alpine 2

6.3 Ovály

Okruhy mající oválový profil jsou charakteristické větším namáháním jedné strany vozu z důvodu koncepce zatáček pouze na jednu stranu. Zatáčky jsou také často velice dlouhé, čili je zapotřebí

počítat s větším namáháním po delší dobu. Nejhůře jsou na tom pneumatiky a zavěšení na vnější straně zatáčky.

Oválové okruhy jsou povrchově tvořeny také asfaltem, ale jsou tak specifické ohledně nastavení a charakteru trati, že mají v práci vlastní podkapitolu.

6.3.1 C-Speedway

Okruh se dá označit jako klasický „americký ovál“ (obrázek 6.8), který je typický pro sérii Nascar případně formulovou Cart sérii. Okruh je celý klopený, nemá tedy klopené pouze zatáčky, ale i rovinné úseky. Je důležité správně nastavit podvozek (tlumiče, pružiny, stabilizátory), aby nepůsobily přehnané síly na pneumatiky na vnější straně zatáčky. V opačném případě reálně hrozí prasknutí pneumatiky. Na oválovém okruhu, kde se rychlost pohybuje velmi vysoko, vnější stranu lemují zed' a vozů je na trati i více jak 30, je tato hrozba daleko nebezpečnější.



Obrázek 6.8 C-Speedway [33]

U techniky PSO se povedlo vylepšit čas na kolo o zhruba půl sekundy (tabulka 6.9), což je na okruhu kde se jede neustále na plný plyn velmi dobré zlepšení.

GA	PSO	HGAPSO	MANUAL
40:24	37:53	42:03	38:16

Tabulka 6.9 - Časy na kolo pro okruh C-Speedway

V rámci nalezeného nastavení u techniky PSO je vidět slušné nastavení převodovky, kdy jednotlivé převodové stupně jsou dostatečně dlouhé pro dosažení maximální rychlosti mimo nejvyššího. Ovšem tato hodnota může snížit právě zmíněnou maximální rychlost. Velmi pěkně jsou nastaveny pružiny a tlumiče. Jejich nastavení se pohybuje ve druhé polovině intervalu a tyto hodnoty zajišťují kompromis mezi dobrou stabilitou v klopených zatáčkách a potřebnou jistotu z důvodu nebezpečí prasknutí pneumatiky. Také nastavení odklonu kol je přizpůsobeno klopené zatáčce, i když zde je nutné přihlížet s určitou rezervou, protože na okruzích s klopenými zatáčkami se spíše nastavují levá a pravá kola než přední a zadní. Další specifikum oválového okruhu je nastavení brzd, kde není zapotřebí hledat žádné složitosti, jelikož se v podstatě nepoužívají.

6.3.2 Michigan Speedway

Oválnový okruh Michigan Speedway (obrázek 6.9) je velmi podobný skutečnému (v Michiganu). Oproti reálnému okruhu jsou zatáčky ostřejší a okruh není (prakticky) klopený. Mírnou zvláštností se dá nazvat umístění boxových stání, která nejsou umístěná u startovní (cílové) čáry. Oproti předchozímu okruhu je ovál Michigan Speedway náročnější na brzdy, ale to je způsobené závodním režimem C-Speedway, kde se prakticky nebrzdí.



Obrázek 6.9 Michigan [33]

Podobně jako u předchozího okruhu i zde je nalezeno (nepatrné) zlepšení. Podle profilu trati by se čekalo výraznější zlepšení spíše u aktuálního okruhu než u předchozího. Ovšem jak již bylo zmíněno veškeré výsledky (pozitivní či negativní) je třeba brát s rezervou.

GA	PSO	HGAPSO	MANUAL
40:03	39:91	38:86	38:97

Tabulka 6.10 - Časy na kolo pro okruh Michigan Speedway

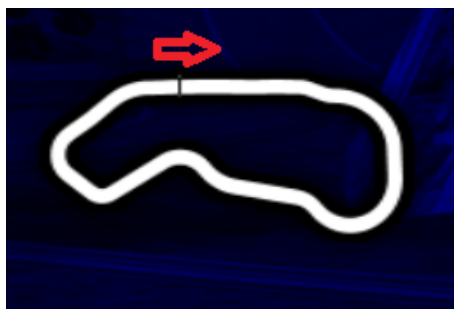
Nastavení nejúspěšnější techniky (HGAPSO) nabízí některé dobře nastavené parametry například nastavení přitlačných křídel, kde jsou nastaveny malé úhly. Slušně je nastavená také světlá výška, kde nejsou tak velké rozdíly mezi přední a zadní částí vozu. Rozumné hodnoty jsou k nalezení v kolonce průměrů brzdových kotoučů. Na druhou stranu nalezené nastavení převodových stupňů nepatří k nejlepším. Dokonce by se dalo tvrdit, že by mohli nastat konstrukční problémy při výrobě dané převodovky.

6.4 Štěrkové okruhy

Štěrkové okruhy se vyznačují především kratší vzdáleností, jsou zhruba o třetinu až polovinu kratší než klasické asfaltové okruhy. Další vlastností je výrazně menší přilnavost. Více informací je uvedeno u vybraných okruhů.

6.4.1 Dirt 1

Okruh Dirt 1 (obrázek 6.10) je na šterkovy okruh poměrně schůdný. Okruh jako celek se nachází na rovinatém území a je dobře vidět na výjezd ze zatáček. Také přilnavost okruhu je relativně slušná, což je způsobeno řekněme středně tvrdým povrchem. Jinak je okruh typicky šterkovy, čili je výrazně kratší než klasický asfaltový. Také je potřeba nastavit převodovku na kratší převody (akcelerace), jelikož na podobných okruzích se k maximální možné rychlosti vozu nikdo nepřiblíží.



Obrázek 6.10 Dirt 1 [33]

Od předchozích zlepšení nyní přichází lepší čas od genetického algoritmu. Zlepšení je v celku slušné cca půl sekundy.

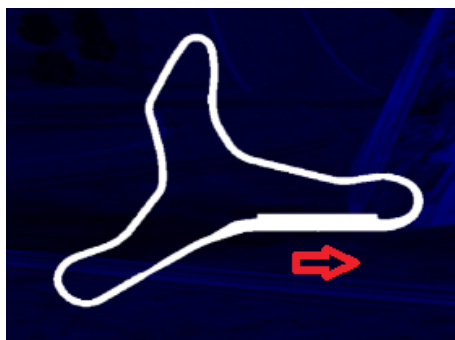
GA	PSO	HGAPSO	MANUAL
33:04	35:61	33:74	33:43

Tabulka 6.11 - Časy na kolo pro okruh Dirt 1

Jako plus se dá označit kratší převod číslo 2 a 3, které zajistí dobrou akceleraci ze zatáček. Podobné ohodnocené kritérium se dá označit i nastavení světlé výšky, která má vyšší hodnotu než u asfaltových okruhů. Ovšem hodnoty nastavení tlumičů a pružin odpovídají (právě) okruhům s asfaltovým povrchem.

6.4.2 Dirt 3

Okruh Dirt 3 (Obrázek 6.11) je náročnější co se týče profilu a převýšení trati. Tento okruh nejvíce připomíná prvky rallye. Je také poměrně dlouhý. První pasáž okruhu neustále stoupá a vyvrcholí ve třetí zatáčce, která se jede přes horizont, což znamená nulovou viditelnost na druhou část zatáčky. Ze třetí zatáčky se sjíždí dolů a tudíž je nutné správně (především dříve) zabrzdit do čtvrté zatáčky. Poslední zálužností je předposlední zatáčka (na obrázku 6.11 v levém dolním rohu), kterou lemuje po vnitřní straně svah, tudíž není vůbec vidět do středu zatáčky natož na výjezd. Naopak velkou výhodou okruhu je poměrně tvrdý podklad, čili přilnavost se dá označit za velmi dobrou.



Obrázek 6.11 Dirt 3 [33]

Okruh patří k delším (co se týče šterkových) a je vidět fakt popsany v kapitole 6.1. Optimálně zajetý čas (uživatel) je kolem minuty a žádný z algoritmů nenalezl lepší čas. Ovšem rozdíly nejsou tak markantní jako u delších okruhů.

GA	PSO	HGAPSO	MANUAL
1:03:07	1:12:47	1:11:12	1:01:26

Tabulka 6.12 Časy na kolo pro okruh Dirt 3

Opět je zapotřebí zmínit správně nalezené nastavení převodovky u genetického algoritmu, kde jednotlivé převody jsou kratší (vede k větší akceleraci). Hodnota rozložení brzd je nastavena mírně na zadní část, může nastat mírná přetáčivost vozu, ale tento fakt nemusí být u aktuální trati negativní. Co je nastavené divoce je světlá výška, přední část vozu je 2x vyšší než zadní. Stejně tvrzení se dá prohlásit o kontrastu nastavení pružin a tlumičů. Pružiny jsou nastaveny jako měkké (velmi správně) a tlumiče jsou nastaveny v přední části na tvrdé a v zadní části na extrémně měkké.

6.4.3 Dirt 5

Okruh Dirt 5 (Obrázek 6.12) je nejnáročnějším okruhem na přilnavost, jelikož okruh je nakypřený, což znamená, že se hodně práší. Tento fakt výrazně zhoršuje viditelnost a dá se označit za největší záludnost okruhu. Okruh se jede po směru hodinových ručiček, tudíž má větší počet pravých zatáček, které se často zavírají. Toto znamená, že vůz bude často klouzat do svodidel na výjezdu ze zatáček. Zmíněná nepříjemnost se eliminuje dřívějším brzděním do zatáček a také nastavením dobré akcelerace. Vlastnost se nejvíce projevuje v zatáčkách číslo 2 a 7.



Obrázek 6.12 Dirt 5 [33]

Okruh Dirt 5 nabízí zlepšení času na kolo u dvou technik. Zlepšení se dá označit za velmi významné. Opět je ale nutné zmínit, že veškeré výsledky je nutné brát s rezervou.

GA	PSO	HGAPSO	MANUAL
27:84	28:44	33:93	32:96

Tabulka 6.13 - Časy na kolo pro okruh Dirt 5

U genetického algoritmu je velmi dobré nastavení převodovky v podobě kratších převodových stupňů. Ovšem u ostatních parametrů nelze vyčíst výrazný posun vpřed, což jen naplňuje spekulace o věrohodnosti simulátoru.

6.4.4 Mixed 1

Okruh Mixed 1 (Obrázek 6.13) je největší výzvou pro nalezení správného nastavení, jelikož je tvořen jak z asfaltového, tak šterkového povrchu. Asfaltový povrch je od předposlední zatáčky po zatáčku číslo 2. Situaci ztěžuje fakt, že šterkový povrch je velmi podobný jako na okruhu Dirt 5, čili jsou dva povrchy s naprosto odlišnými vlastnostmi. Na jedné straně povrch s výbornou přilnavostí a na druhé povrch, který má přilnavost minimální. Minimální přilnavost se projeví především v zatáčce číslo 4, do které se najíždí z nejvyšší rychlosti na šterkové části a je tvořena v podstatě dvojí zatáčkou, tudíž se zavírá. Vůz se zde bude tlačit na vnější stranu do svodidel. Podobně to bude v zatáčce číslo 7 (první čistá zatáčka o 180 stupňů). Zde bych se přikláněl spíše k nastavení na šterkový povrch, především co se týká převodovky. Ohledně nastavení podvozku by bylo možno nastavit (mírně) na asfalt, jelikož šterkový povrch není tvořen nerovnostmi a dalo by se získat cenné desetiny vteřiny.



Obrázek 6.13 Mixed 1 [33]

Relativně kratší okruh, tudíž nepřekonání zajetého času je menším zklamáním, dá se ovšem přisoudit velkou náročností na nastavení (okruh je tvořen ze dvou povrchů viz výše).

Jak ukazuje tabulka 6.14 nejlepší čas byl dosažen pomocí techniky PSO. Celkově je nastavení přikloněno spíše šterkovému povrchu, čemuž napovídá světlá výška vozu nebo nastavení pružin a převodovky. Průměry brzdových kotoučů jsou v pořádku stejně tak rozložení brzd, které je nastaveno

více na přední část. Ovšem nastavení předních tlumičů se vymyká normálu, jelikož jejich nastavení je voleno výrazně na povrch asfaltový bez větších nerovností.

GA	PSO	HGAPSO	MANUAL
50:73	47:14	48:61	45:39

Tabulka 6.14 - Časy na kolo pro okruh Mixed 1

6.5 Shrnutí

Kapitola shrnula nejlepší výsledky vybraných okruhů. U kratších okruhů je vidět zlepšení časů, naopak u delších byl velký problém se vůbec zajetým časům přiblížit. V kapitole 6.1 jsou uvedeny nejdůležitější poznatky práce, následující kapitoly popisují testované tratě a ukazují výsledky algoritmů. Zároveň každá kapitola upozorňuje na největší zajímavosti nalezeného nastavení.

U většiny okruhů je popsáno nalezené nastavení převodovky, z důvodu náročnosti nastavení. Jedná se o 5 parametrů, které na sebe úzce navazují. U některých řešení je velice pravděpodobné, že by bylo složité vůbec převodovku vyrobit a také, jak moc by byla pohodlná pro jezdce. Jelikož pokud z jednoho stupně při konstantním zrychlování přeřadí během 3 sekund a z následujícího by musel přeřadit během půl sekundy⁹, nebyl by schopen se plně soustředit na jízdu a zajet optimální kolo.

Problém převodovky nejlépe vystihuje zrádnost hledání nastavení vozu přes „přírodní“ optimalizační techniky a nedůvěryhodnost programu TORCS. Na druhou stranu i nejlepší simulátor se nemůže rovnat (alespoň prozatím) testování na skutečném okruhu.

⁹ Časové okamžiky jsou pouze odhadem, jde spíše o poměr těchto časů.

7 Závěr

V práci jsou popsány možné způsoby řešení problému nastavení parametrů závodního vozu za pomoci optimalizačních metod inspirovaných přírodou.

Jako optimalizační techniky byly zvoleny genetické algoritmy, optimalizace hejnem částic (PSO) a jejich spojení do hybridní techniky (HGAPSO). Kapitola 2 se věnuje zmíněným technikám a jejich jednotlivým parametrům.

V následující kapitole je rozvedeno nastavení parametrů závodního vozu. Snahou kapitoly popisující nastavení parametrů je dodržení pořadí nastavení u reálného závodního vozu. Druhou částí kapitoly je popsání simulátoru TORCS přes principy, komunikaci a představení (možných) nastavitelných parametrů.

Po uvedení návrhové a implementační části se práce ubírá k představení testovacích okruhů a popisu jednotlivých výsledků optimalizačních technik.

Veškeré výsledky jsou porovnány s bezchybně zajetými koly v programu TORCS s totožným závodním vozem. V rámci diplomové práce byly získány hodnoty respondentů, kteří jsou zainteresováni v oblasti strojírenství. Avšak výsledky těchto nastavení byly o několik tříd horší, čili nebyly v práci dále uvažovány. Skutečnost špatných výsledků respondentů bych přisuzoval spíše nedokonalosti simulátoru, jelikož časy kol, které byly vráceny, odpovídaly možnostem obyčejných dopravních vozů. Rozdíl se pohybovaly od 30 sekund do 2 minut na kolo, což jsou ve světě závodních vozů nepředstavitelné rozdíly. Tvrzení také podporují určité výsledky simulátoru v rámci optimalizačních technik, kde navrácené hodnoty příliš neodpovídají realitě. Vše dokládá uvažované nastavení převodovky, které bylo rozebíráno prakticky u všech testovacích tratí. V určitých případech by byl problém již v oblasti konstrukce. V jiných případech by byla převodovka prakticky nepoužitelná, protože rozdíly mezi jednotlivými převodovými stupni jsou nezdravě velké. Jezdec by si musel zvykat na nepravidelnost mezi různými převodovými stupni.

Nalezení ideálního nastavení také brzdila doba simulace programu TORCS, kdy se musel volit kompromis mezi vyhodnocením aktuálních výsledků a délkou běhu optimalizačních algoritmů. Tato kombinace zvláště u delších okruhů znemožňovala překonat zajetý čas bezchybného kola. U kratších okruhů zmíněné omezení nedělalo tak výrazné problémy a podařilo se nalézt slušné nastavení závodního vozu. Všechny tři zvolené optimalizační techniky se podílely na vylepšení časů na kolo, převážně u kratších okruhů a tudíž se povedlo nalézt lepší nastavení. Lepší výsledky byly dosaženy především pomocí genetického algoritmu a optimalizace hejnem částic.

Literatura

- [1] KENNEDY, James a Russell C. EBERHART. *Swarm Intelligence*. 2001. San Diego, CA: Academic Press, 2001. ISBN 978-1-55860-595-4.
- [2] BRANKE, Jurgen. *Evolutionary Optimization in Dynamic Environments*. 2002. Norwell, Massachusetts USA: Kluwer Academic Publishers Group, 2002. ISBN 0-7923-7631-5.
- [3] FANG, Luping, Pan CHEN a Shihua LIU. Particle Swarm Optimization with Simulated Annealing for TSP. 2007, 5. Dostupné z: <http://www.wseas.us/e-library/conferences/2007corfu/papers/540-365.pdf>
- [4] SHI, X.H., Y.C. LIANG, H.P. LEE, C. LU a Q.X. WANG. Particle swarm optimization-based algorithms for TSP and generalized TSP. 2007, 8. Dostupné z: [http://ccst.jlu.edu.cn/ie/publications/IPL-2007\(103\)169-176.pdf](http://ccst.jlu.edu.cn/ie/publications/IPL-2007(103)169-176.pdf)
- [5] Schwarz, J.; Sekanina, L. *Studijní opora předmětu EVO*. Brno VUT-FIT, 2006
- [6] Particle Swarm Optimization in Wikipedia: the free encyclopedia [online]. Dostupný z: http://en.wikipedia.org/wiki/Particle_swarm_optimization
- [7] SCHUTTE, Jaco F. *The Particle Swarm Optimization Algorithm* [online]. 2005. 2005, s. 33 [cit. 2011-12-27]. Dostupný z: http://www.mae.ufl.edu/haftka/stropt/Lectures/PSO_introduction.pdf
- [8] POLI, KENNEDY a Tim BLACKWELL. *Particle Swarm Optimization* [online]. 2007, s. 30 [cit. 2011-12-27]. Dostupný z: <http://natcomp.liacs.nl/NC/slides/psa.pdf>
- [9] KENNEDY, James a Russell EBERHART. Particle Swarm Optimization. s 7. Dostupné z: http://chern.ie.nthu.edu.tw/gen/_The_Swarm-original_paper.pdf
- [10] POLI, Riccardo, James KENNEDY a Tim BLACKWELL. Particle swarm optimization. 2007, s 25. Dostupné z: <http://dces.essex.ac.uk/staff/rpoli/papers/PoliKennedyBlackwellSI2007.pdf>
- [11] Hejtmánek, P. *Nastavení podvozku závodních automobilů*. Brno VUT-FSI, 2007, s 26
- [12] Bjorkman, R. Základní nastavení vozu. 2008. [online]. Dostupný z: http://cz.lfsmanual.net/wiki/Z%C3%A1kladn%C3%AD_nastaven%C3%AD_vozu
- [13] Smith, B. Pokročilejší nastavení vozu. 2009. [online]. Dostupný z: http://cz.lfsmanual.net/wiki/Pokro%C4%8Dilej%C5%A1%C3%AD_nastaven%C3%AD_vozu
- [14] Cardamone, L.; Loiacono, D.; Lanzi, P. *Simulated Car Raving Championship*. Software manual. Politecnico di Milano, 2009
- [15] Cardamone, L.; Loiacono, D.; Lanzi, P. *Car Setup Optimization Competition*. Software manual. Politecnico di Milano, 2010

- [16] Jesse, F. The Live for Speed S2 Alpha Manual. 2005. [online]. Dostupný z: http://www.4players.de/rendersite.php?LAYOUT=download_start&world=ifs&KAT_PARENT=121&DOWNLOADID=44692
- [17] Formula one. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.formula1.com/>
- [18] World Rally Championship. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.wrc.com/>
- [19] World Touring Car Championship. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.fiawtcc.com/>
- [20] Portál f1. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.formule.cz/formule/>
- [21] Computational Intelligence in Games, Milano 2011 [online]; Dostupný z WWW: <http://cig.dei.polimi.it/?page_id=79>
- [22] Genetic and Evolutionary computation conference – GECCO. [online]. 2011. Dostupný z WWW: <<http://www.sigevo.org/gecco-2012/>>
- [23] Cellular automaton in Wikipedia: the free encyclopedia [online]. Dostupný z: http://en.wikipedia.org/wiki/Cellular_automaton
- [24] Juang Ch. F. A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design, 2004, 10. Dostupné z: https://cours.etsmtl.ca/sys843/pdf/Juang_IEEE-Tr-SMC_2004.pdf
- [25] Okruh v Monaku. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.fl.sport24.pl/pliki/image/formula1/tor/monako.jpg>
- [26] Okruh v Monze. [online]. [cit. 2011-12-27]. Dostupné z: http://fl.automobile-magazine.fr/circuits/gp_italie.jpg
- [27] Vůz F1. [online]. [cit. 2011-12-27]. Dostupné z: <http://www.race-f1.com/content/articles/601.jpg>
- [28] Vůz wrc. [online]. [cit. 2011-12-27]. Dostupné z: <http://cdn-www.rsportscars.com/images/ford/2008-ford-focus-rs-wrc/focuswrc08.jpg>
- [29] Vůz wtcc. [online]. [cit. 2011-12-27]. Dostupné z: http://www.tamiyausa.com/images/product/300/58385/header_58385.jpg
- [30] SCHWARZ, Josef. *Genetické algoritmy* [online]. 2006, s. 39 [cit. 2011-12-28]. Dostupný z: https://www.fit.vutbr.cz/study/courses/EVO/private/VYUKA/prednasky_2011/P1/GA_sato.pps
- [31] Hejno ptáků. [online]. [cit. 2011-12-28]. Dostupné z: <http://www.ireceptar.cz/res/data/100/012278.jpg>
- [32] Hejno ryb. [online]. [cit. 2011-12-28]. Dostupné z: http://4.bp.blogspot.com/_ek9gFHiRWsw/TJCWjHYtDbI/AAAAAAAAAD14/Wu0rIzMQIQ/s1600/Hejno.jpg
- [33] Obrázky použité z programu TORCS
- [34] Webové stránky projektu TORCS [cit. 2012-02-03]. Dostupné z: <http://torcs.sourceforge.net/>

Příloha 1 - Zprovoznění projektu

Kapitola popisuje veškeré kroky, které jsou nutné pro získání zdrojových kódů, programu TORCS a použití implementační části této práce. Veškerý postup je uveden jak pro systém Windows, tak i pro systém Linux. Podobný postup se nachází v [15].

Windows

Nejdříve je nutné stáhnout program TORCS z [34] nebo je k nalezení na příloženém cd (složka win-install). Instalace programu je jednoduchá, jedná se pouze o zvolení cílové složky programu. Dále je nutné stáhnout zdrojové kódy projektu CIG z [21] nebo opět na příloženém cd (složka win-install). Obsah staženého archivu je nutné zkopírovat do složky, kde je nainstalovaný program TORCS. Proces kopírování bude přerušeno s dotazem ohledně přepsání souborů, zde je nutné zvolit možnost přepsání všech souborů. Jako poslední je zapotřebí stáhnout klientskou aplikaci v jazyce Java z [21] nebo ve složce win-install na cd. Po rozbalení archivu je důležité zaměnit soubor SimpleGA.java za libovolný s danou technikou. Poté se projekt přeloží například vytvořením (a přeložením) projektu v Netbeans nebo Eclipse.

Jako první se spouští server (wtorcs.exe), kde se zvolí:

Race → Practice → New race

Následně se spustí klientská část a ihned poté se začnou vypisovat výsledky. Důležité jsou především první dva sloupce, kde jsou výpisy nejlepšího kola aktuální generace a nejlepšího kola vůbec.

Linux

Pro systém Linux platí velmi podobný postup. Nejdříve se stáhne program (archiv) TORCS z [34] nebo je archiv k nalezení na cd (složka linux-install). Po rozbalení archivu se přejde do složky programu (*cd torcs-1.3.1*) a pomocí skriptu *./configure* se zjistí, které knihovny jsou potřeba k instalaci a spuštění projektu. Pokud skript zjistí absenci určité knihovny, je vždy zapotřebí (po úspěšné instalaci knihovny) spustit skript opětovně.

Opět je nutné stáhnout archiv projektu CIG [21] nebo je archiv k nalezení na cd (složka linux-install). Obsah archivu se rozbálí do složky projektu TORCS. Přejde se do složky optpatch (*cd optpatch*) a spustí se skript příkazem *sh do_patch.sh*.

Další částí je přeložení a instalace programu, což se zajistí 3 příkazy:

\$ make

\$ make install

\$ make datainstall

Od této chvíle je postup totožný s postupem ve Windows, čili nejdříve stažení klientské části, záměna souboru SimpleGA.java za soubor s vybranou technikou, vytvoření projektu například v Eclipse, zkompilování a spuštění projektu. Poté se spustí program TORCS příkazem *torcs* a zvolí se:

Race → Practice → New race

Pokud chceme změnit testovanou trať, tak při zvolení možnosti *Practice* místo *New race* se zvolí *Configure race*, kde se daný okruh zvolí a poté je nutné se proklikat přes nastavení jezdce a počtu kol. Jezdec je primárně nastaven *optserver 1*, což je v pořádku a počet kol na 2000. Tento počet je dostačující. Pokud by byl nastaven na menší je vhodné počet zvýšit právě na zmíněnou hodnotu.

Příloha 2 – Nalezená nastavení

Příloha obsahuje tabulky s nalezenými nastaveními v podobě tabulek. Každý okruh má vlastní tabulku, která má pro lepší orientaci v popisku název daného okruhu. Jednotlivé hodnoty jsou záměrně uvedeny na více desetinných míst pro lepší přesnost v simulátoru. Čísla odpovídají parametrům uvedeným v kapitole 3.4.3.

	GA	PSO	HGAPSO
1	0.879003541225	4.26094451347	2.5
2	4.56516954428	2.8977362632	1.15603905279
3	4.38416028044	4.38651060257	2.5
4	1.1136841125	1.17866642622	1.88476149983
5	1.21555114671	3.52259296973	2.5
6	8.23648214774	17.9209582954	7.48584862796
7	5.48890883677	5.44846260882	10.3560673775
8	0.360680385205	0.544555919679	0.322944615435
9	77492.6322609	29051.2830348	113357.779842
10	4964.38768136	2014.64497279	3509.12286422
11	1618.87511307	2544.63958897	2383.56436304
12	240.919560928	225.88634311	163.719999562
13	0.124096452127	3.24446514799	4.10665591612
14	1.12356135603	0.275212982388	-1.0
15	136.239440523	218.259616049	124.929230131
16	-3.89512653037	-3.44512353895	-2.77116009785
17	7448.75302809	1614.46881926	8256.15621475
18	0.0730792986837	0.040008909184	0.1
19	6395.11177122	108.256487091	7799.18338587
20	0.150936955979	0.183967076758	0.0786809501018
21	353.248874143	206.277509024	184.891969182
22	140.244676425	238.959763988	240.0

Tabulka 1 - Výsledky pro okruh Forca

	GA	PSO	HGAPSO
1	3.83165175484	3.64543184573	2.5
2	1.7209272916	2.29346190279	2.03198050656
3	2.40183373278	2.18410801918	1.07177487587
4	0.844863962629	1.84892896789	1.67126153155
5	2.27874278998	0.938264086186	2.5
6	1.95475874681	0.272830881268	0.882477692339
7	4.31663373112	4.79923223425	6.0
8	0.34011042787	0.320526205362	0.522468654893
9	134503.347387	93519.5753372	75050.0
10	580.30525082	1299.34851176	2500.0
11	847.381999284	4582.68408271	2500.0
12	169.917928334	290.360737102	246.025478112
13	1.13464575043	-2.39565138968	-3.33530606294
14	0.338832812573	-4.23550534285	-0.795109844605
15	218.193349945	254.191353769	200.0
16	-4.36458501256	-3.15899016494	-3.5
17	8843.72464461	4035.94317612	5000.0
18	0.124866112896	0.0535981908503	0.196175618005
19	897.997432366	2122.25120896	1373.33822228
20	0.0509999535582	0.0565139506258	0.1
21	217.453258596	275.607888692	240.0
22	309.107406997	100.110652655	240.0

Tabulka 2 - Výsledky pro okruh CG1

	GA	PSO	HGAPSO
1	3.94639095748	3.22372547277	2.5
2	2.26641550617	3.772661013	3.80409542128
3	1.59012935272	1.32934701366	3.35551738593
4	0.781520040705	1.54305157391	1.48870997517
5	3.55737127577	3.27346912599	0.532956372431
6	7.30514946082	7.81418673198	9.0
7	1.33206859207	2.31528210001	6.0
8	0.413629756855	0.540514114654	0.590067702039
9	13501.4353479	91707.2084603	26247.0380503
10	1552.97032556	3614.04655392	3549.72365088
11	2843.02451109	2317.18672876	2500.0
12	231.29312008	240.79557783	200.0
13	0.932196932986	3.40506413692	3.20040223195
14	-1.69404411109	-0.729606344464	-2.01088782205
15	124.757785222	137.793901868	110.807606366
16	-3.09327074848	-3.72553466371	-4.70422935337
17	2993.36892174	2681.9407672	4626.6576737
18	0.164201609799	0.0519187715414	0.1
19	8958.07785008	3233.19456614	5135.04385975
20	0.123562305503	0.159953159917	0.0717780012049
21	287.178604126	320.726790575	358.339493867
22	183.676296596	259.881918587	275.28840909

Tabulka 3 - Výsledky pro okruh CG2

	GA	PSO	HGAPSO
1	4.41560714739	3.97728399223	3.39189856369
2	0.897710030813	3.36281205213	2.68117490342
3	4.83617955162	2.88028729146	2.50062100575
4	2.02993613674	0.0733864008242	2.36883619187
5	3.08941619364	0.96545708288	2.13908688853
6	3.45236920978	1.88703169897	9.81220496539
7	1.62225355153	3.65673715562	5.58034022377
8	0.563709619041	0.311122885391	0.42152391964
9	140922.651993	60354.3559372	102911.264088
10	2113.61481835	3664.21577121	1577.56351487
11	1340.58905099	3881.94769351	2054.59760814
12	176.475022046	228.495952923	187.248141374
13	-1.91150570014	-3.88263599488	1.19934371067
14	2.52975218121	-3.26051631762	-3.38889201607
15	160.19139781	177.192430613	198.409528355
16	-2.83865225287	-4.35853822149	-2.53428056884
17	687.185139147	6397.34381904	6537.80667596
18	0.121104085638	0.0952336567212	0.0946412744206
19	9643.43385829	499.751376373	3069.83929009
20	0.141662493009	0.163053684071	0.0963092296663
21	125.436860389	337.380203875	289.450743785
22	341.500302825	311.158106047	135.458653431

Tabulka 4 - Výsledky pro okruh Olethros

	GA	PSO	HGAPSO
1	3.05242633099	4.33585893041	3.99792398867
2	1.39162787482	3.29834599558	2.5
3	3.58669894949	3.67686085723	3.52468120964
4	0.728929941845	4.31597618737	2.19568564218
5	4.82957332877	0.649273948396	0.711773323334
6	2.17427678607	5.56250446462	14.4715723332
7	8.2699050015	4.71206374717	6.7857397889
8	0.62648589853	0.65358241223	0.340724281073
9	90941.9482902	112636.769187	78838.3489763
10	514.653794942	4606.7938988	1635.37206628
11	3586.59100855	2836.74598453	2812.49305924
12	262.558432458	166.6938464	135.128831995
13	2.1218552013	1.15624347688	1.12171021147
14	-4.37004485093	1.56592740845	-0.915477226474
15	173.291384368	184.85772877	228.020157302
16	-2.39177391182	-2.89002778978	-2.39153107717
17	6760.51346532	7726.76990298	6160.60761052
18	0.000894894493793	0.111271707606	0.0750456624749
19	3081.84073178	3460.14900132	2611.43139522
20	0.158441963466	0.128307625592	0.0789074114782
21	270.894499239	200.864410724	185.21112978
22	295.606709487	163.995675076	301.917468224

Tabulka 5 - Výsledky pro okruh Street 1

	GA	PSO	HGAPSO
1	1.07355197797	3.31802808388	0.935394935438
2	0.872269992358	1.66387355627	2.80218891748
3	1.51756785609	3.24030132395	2.84606025266
4	1.02456287258	4.41119064646	3.20596407171
5	2.43198086797	2.51744069385	3.31802149317
6	5.97580424098	10.1738042613	1.24953895689
7	11.5814187436	0.760099569471	7.39533478808
8	0.566649158638	0.378144371338	0.455626828331
9	62557.6866388	140108.558667	132418.660008
10	929.713753996	4439.02257003	3090.25801134
11	4391.13384726	3733.46461348	2506.26794565
12	298.444787554	165.869948719	140.789077789
13	1.6910299986	0.216525134206	-0.874237349385
14	2.48599716478	-2.96708875072	-1.15604830678
15	148.85259027	243.2282932	205.334003276
16	-2.49175008625	-2.4614272401	-4.2473642912
17	1569.54576338	6973.21006984	2193.42988864
18	0.0611107554609	0.0912577849468	0.103641873106
19	7615.28401153	2902.26942193	4232.98609982
20	0.176321710069	0.0733415557497	0.112473850884
21	210.332313844	378.746678657	339.06690971
22	250.163396945	140.061398554	198.16161172

Tabulka 6 - Výsledky pro okruh Wheel 2

	GA	PSO	HGAPSO
1	4.97039112178	0.924257182323	2.00914223511
2	2.5798206334	3.4139366088	3.49840169345
3	2.40192676202	3.89976200404	1.09268807792
4	1.83306570538	4.62076240965	3.80921713794
5	1.1091582786	2.86513013575	3.57671591657
6	11.1938988791	2.52154159929	9.21684703899
7	0.289631073704	6.49248029338	0.67340824316
8	0.547549699347	0.550778671924	0.659350905539
9	7845.79261853	72992.6082403	141796.141877
10	1070.65002397	4500.30489641	589.649997089
11	2214.62875125	4033.11832465	2640.82208637
12	156.97691885	298.821646405	105.658055546
13	0.216700213238	-0.950429175262	4.10626157352
14	-4.36069474481	-4.27139825609	-4.35518529398
15	102.469169955	104.972953926	112.237515051
16	-3.12340764628	-3.0225277737	-4.16774411142
17	5794.46923597	2341.53944077	6006.64145964
18	0.0473063843109	0.116821895081	0.101573620934
19	3778.57695389	7533.66546897	4709.07309937
20	0.0590561855938	0.198772639436	0.192899306974
21	119.375161486	186.823889197	136.60733932
22	248.949117436	108.087883936	127.979032657

Tabulka 7 - Výsledky pro okruh Alpine 2

	GA	PSO	HGAPSO
1	3.87520101694	3.34379934834	2.5
2	2.65351333832	4.58408635678	0.823124858099
3	2.46316803514	4.76720095357	2.59261483788
4	4.88207439697	4.55315856748	0.215150695222
5	0.831239411972	0.428974718662	3.58830833728
6	5.20894427408	16.9978740357	14.8239074776
7	8.75533099131	2.56269119841	11.2948759287
8	0.419029397092	0.460345230861	0.688966830764
9	131336.510602	54151.5932234	75050.0
10	1675.57522622	3715.9857864	2486.93820069
11	603.845098756	3805.45722242	3810.80204264
12	214.223963493	138.412356361	200.0
13	0.622028279807	-1.11675232003	-1.33857004245
14	-0.207374532496	-4.72263445175	2.04171811261
15	116.425596213	119.552803463	206.155326616
16	-3.49176778874	-3.08863425088	-4.4960251877
17	733.332150571	8534.45034372	3844.73227284
18	0.0993004327936	0.0870237671899	0.16768891617
19	3573.61340024	3877.77480736	5000.0
20	0.00337799624387	0.0317137886768	0.192065272533
21	175.32627985	342.262992471	336.070494499
22	255.432924846	329.145519596	240.0

Tabulka 8 - Výsledky pro okruh C-Speedway

	GA	PSO	HGAPSO
1	0.737564930211	2.5	0.90739815187
2	0.0641493495367	2.5	3.7465242449
3	4.84550957165	2.5	2.5
4	4.30342347161	4.11025285188	0.889435609323
5	0.0455880090515	2.5	1.71779605036
6	1.65502956655	9.0	9.09389072976
7	7.68539660604	6.0	6.0
8	0.584147900602	0.613107679723	0.424812528661
9	107699.497374	75050.0	62526.4317317
10	161.555420807	3968.22053876	769.290364505
11	2488.22629617	2792.952932	1978.14855705
12	287.411012476	291.117739929	195.483132671
13	-1.7892061881	-0.923313015653	-0.595133375903
14	0.419810391304	-1.0	-1.0
15	105.170050706	200.0	200.0
16	-2.41305177578	-3.5	-3.54566964043
17	4547.15320213	5000.0	6269.83495222
18	0.0123147469615	0.1	0.1
19	9076.91417625	5000.0	1639.57399873
20	0.169650960131	0.1	0.0549190573301
21	192.006520036	240.0	323.885762082
22	195.011055008	240.0	240.0

Tabulka 9 - Výsledky pro okruh Michigan Speedway

	GA	PSO	HGAPSO
1	3.04268815542	4.61952214678	1.80302620468
2	1.47379738241	3.27134830974	1.14390375646
3	4.44172648155	1.84799920905	1.45083115177
4	4.7451871771	4.60875431335	3.96011353882
5	4.37371249446	1.38897330998	4.18636994411
6	4.68953651391	11.5299275897	4.24071712111
7	8.6860375678	9.67023806485	10.0890233549
8	0.323335519406	0.306243942995	0.67279225062
9	44020.5093071	121844.144366	30896.5184635
10	3830.82982578	4053.0750522	2883.87047397
11	534.062261642	2605.34482682	2739.66239434
12	231.482281495	228.371549525	174.022041801
13	3.11636748847	3.83159315647	0.998599206733
14	1.9508164	-2.21610611189	2.94058000933
15	217.471079744	238.46159974	143.528610127
16	-4.52793128409	-4.97219561462	-3.5520482695
17	5197.93900157	5983.20405608	4229.03510905
18	0.0116768328847	0.14238774519	0.0589781872331
19	3195.48941925	2623.30233907	2112.08788226
20	0.152175022645	0.173193825018	0.184185794169
21	352.393332255	189.122744791	167.380326038
22	179.669048477	243.26120338	281.82420177

Tabulka 10 - Výsledky pro okruh Dirt 1

	GA	PSO	HGAPSO
1	1.66897344136	4.31742544099	3.96692552119
2	3.593971705	3.28839059339	4.10579348971
3	4.79556616882	4.67968369358	3.15204517148
4	1.17511820974	3.4889394264	3.42967174247
5	2.96951512415	2.72882360744	1.23238427424
6	10.1269050823	15.5273898186	5.67456175769
7	8.96325203934	6.33916457584	6.03914512442
8	0.421151342126	0.43602566644	0.421636472121
9	142816.889286	17832.7100654	41933.1924237
10	1294.82200504	1981.5535933	2332.30729723
11	1771.02239905	475.645606588	2471.0975253
12	292.624374212	208.351352538	153.488514144
13	1.31255697783	-4.359651462	-0.624416119292
14	-1.98936930515	0.368710504346	-1.98554887555
15	104.045786499	229.629430574	200.81226204
16	-3.95149098792	-2.1107355954	-4.5401825675
17	6918.3762434	8553.64153324	8141.00929777
18	0.14098438901	0.155728676414	0.0293470233976
19	536.640906789	9207.33725311	5781.58472973
20	0.111760872761	0.13528110731	0.0907394597328
21	195.606589555	207.652193448	315.619722301
22	348.530434118	341.248582761	290.987379321

Tabulka 11 - Výsledky pro okruh Dirt 3

	GA	PSO	HGAPSO
1	3.03388553227	4.16916815546	2.5
2	3.3462043385	2.1731588785	2.5
3	2.1045905272	0.99597555555	2.5
4	1.53085309713	1.52775655438	2.5
5	1.41058081209	2.47059184869	2.5
6	7.72895777191	13.6482896585	0.230316837405
7	0.293845011267	1.02336485408	6.0
8	0.56836995061	0.367786415193	0.5
9	23250.5421342	63220.6209524	75050.0
10	4100.66175525	816.657009446	4564.63633519
11	4084.61178225	1032.64494211	169.511560713
12	276.160196942	160.180650365	203.578204376
13	-1.89836973644	-2.32835728435	-3.66444053892
14	-0.552386724942	-3.44637753559	-1.0
15	149.540791353	116.769838204	200.0
16	-2.9081864933	-2.64607434844	-3.5
17	3922.70136962	7705.55816454	5000.0
18	0.0771155274476	0.087378546806	0.0564112014768
19	238.336992376	4232.53158031	6057.91435731
20	0.0467944381762	0.071472925054	0.1
21	130.832152863	344.967709302	208.774455369
22	178.099951187	329.186278594	224.112405402

Tabulka 12 - Výsledky pro okruh Dirt 5

	GA	PSO	HGAPSO
1	4.44855303734	4.32444632401	2.62071998832
2	2.63240394369	2.36150667908	2.5
3	1.10449288765	4.32755605816	4.36332514691
4	2.78043018271	0.803542050115	1.18969329897
5	0.682457332419	2.23197030977	2.5
6	10.4390338146	8.16557185653	9.0
7	1.76025689638	8.77534334904	9.48658528144
8	0.658318750162	0.474672564348	0.5
9	80065.5866986	111071.045518	75050.0
10	4154.26151867	3682.06060962	4138.25151744
11	4913.03147316	1754.98938863	2500.0
12	235.904759492	278.597161661	101.593861781
13	2.15033302916	-2.06107397433	1.77253219597
14	0.211973700558	2.71984833025	2.87596806357
15	214.562029957	121.602154144	200.0
16	-3.75416919251	-2.10158852568	-3.5
17	8729.47395644	3685.52813808	813.207745503
18	0.0398236733043	0.134892422197	0.00359448517544
19	254.178597111	1368.05636146	5000.0
20	0.139750727832	0.0620743022647	0.1
21	135.247492797	302.24006673	104.125727043
22	147.664421766	213.232711748	240.0

Tabulka 13 - Výsledky pro okruh Mixed 1

Seznam příloh

Příloha 1 – Zprovoznění projektu

Příloha 2 – Nalezená nastavení

Příloha 3 – CD