

# DARP manuál

## Úvod

Program DARP je určen k řešení a testování dynamického problému rozvozu, anglicky *Dial A Ride Problem*, což je rozšířená varianta problému obchodního cestujícího. Základním vstupem seznam vozidel a v průběhu času přibývajících objednávek.

V této variantě může každé vozidlo jeden čas obsluhovat nejvýše jednu objednávku, kterou je třeba doručit do určitého času.

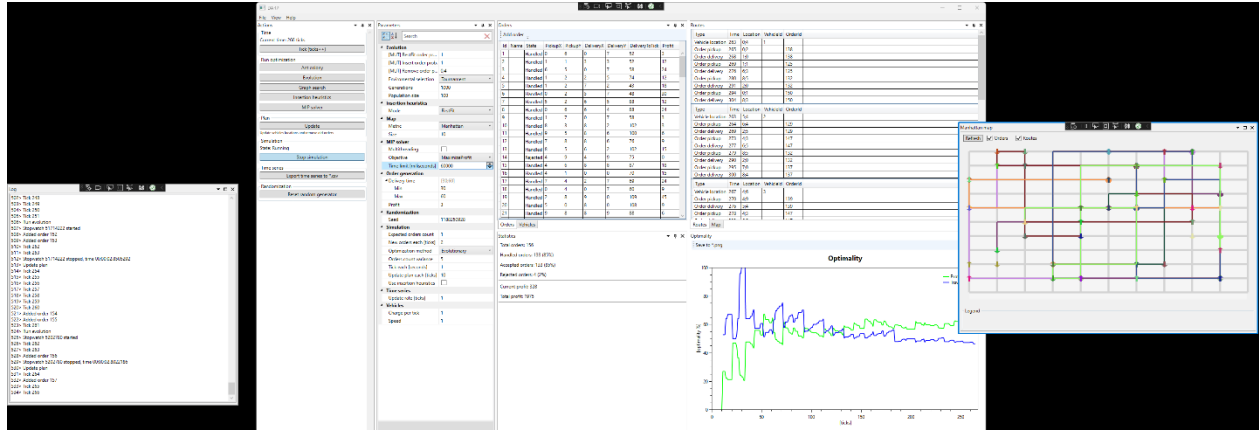
Program nabízí několik optimalizačních metod a jejich výstup je zobrazen jednak v seznamu naplánovaných tras a jednak jako graficky na mapě. Také jsou k dispozici statistiky, kde je možné sledovat různé ukazatele a jejich závislost na vstupních parametrech.

## Obsah

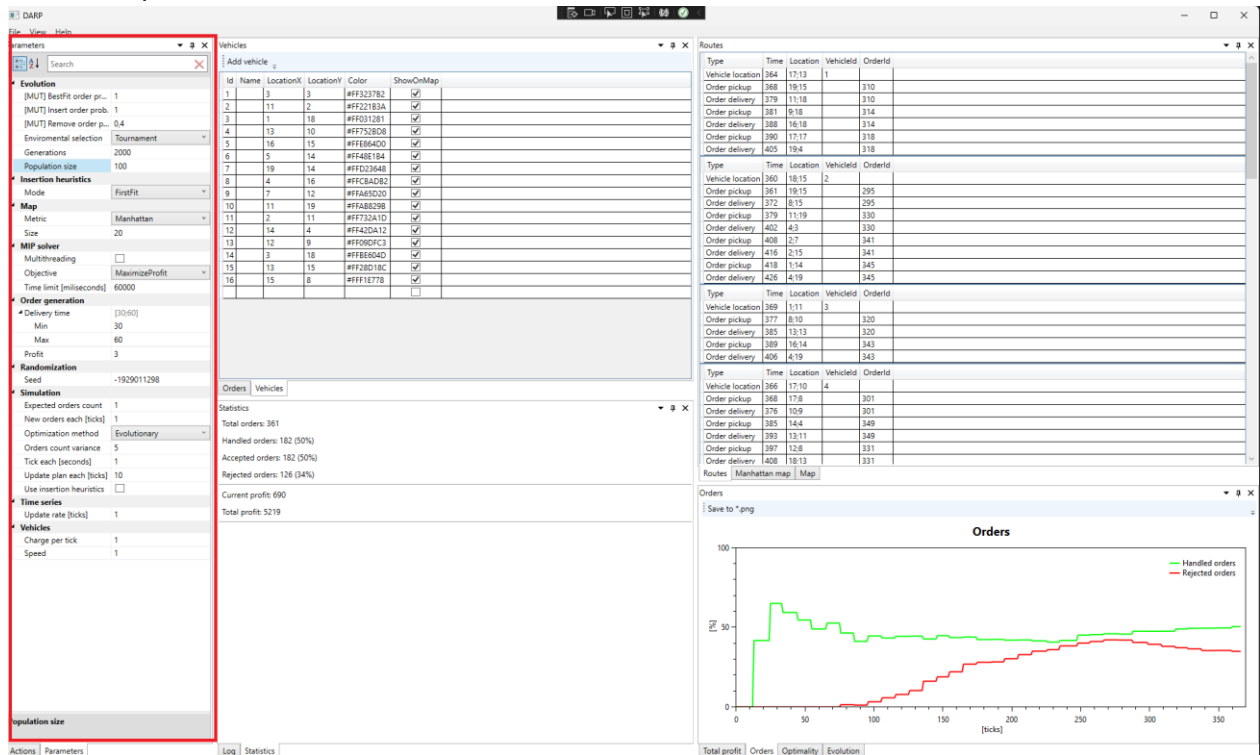
Úvod .....	1
Přehled UI .....	2
Parametry .....	2
Akce .....	3
Objednávky .....	4
Vozidla .....	5
Log .....	6
Naplánované trasy .....	6
Mapa (Manhattan) .....	7
Statistiky .....	8
Optimalizační metody .....	9
Insertion heuristics .....	9
Mixed integer programming .....	9
Evoluční metoda .....	9
Ant colonies .....	10
Kombinované řešení .....	10
Příklady .....	10
Jednorázová optimalizace .....	10
Simulace .....	10
Testování .....	11

# Přehled UI

UI je členeno do menších celků, se kterými lze pracovat jako s klasickými okny nebo je dokovat v rámci hlavního okna. Program si pamatuje poslední rozložení prvků a načte jej při dalším spuštění.

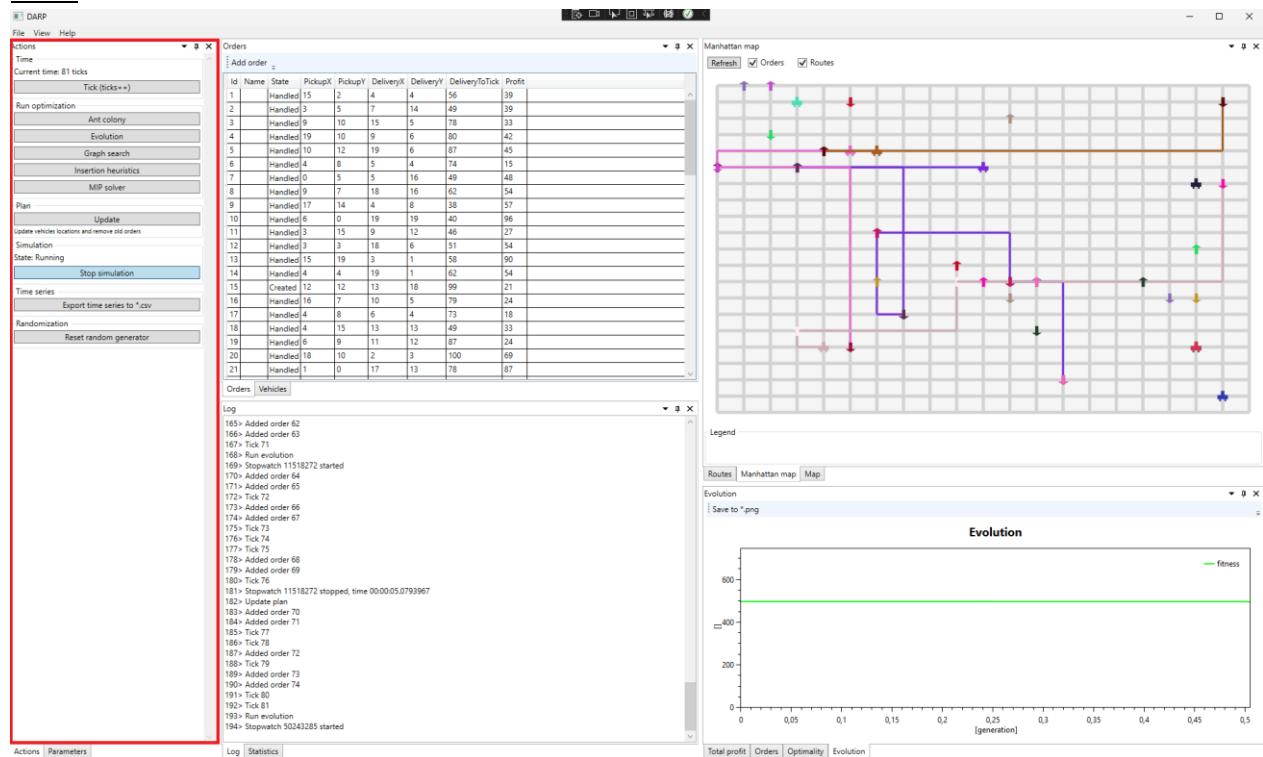


## Parametry



Seznam parametrů, které ovlivňují a určují chování jednotlivých částí programu. Jsou rozdělené do sekcí a jejich změna se projeví vždy okamžitě.

# Akce



Panel akcí nabízí všechny zásadní akce programu. Níže jsou popsány jednotlivé oddíly.

## Time

Zobrazuje aktuální čas v kontextu aplikace. Čas se počítá v jednotkách tiků, při výchozím nastavení odpovídá jeden tik jedné sekundě. V čase se lze posouvat pouze v před stisknutím tlačítka „Tick“.

## Run optimization

Nabídka jednotlivých optimalizačních metod. Kliknutím na tlačítko jednorázově spustí vybranou optimalizační metodu na aktuálním seznamu objednávek ve stavu Created a se zadanými vozidly. Takto lze testovat jednotlivé metody na statickém vstupu.

## Plan

Tlačítko Update aktualizuje plán vůči aktuálnímu času (počtu tiků) a to tím způsobem, že přesune vozidla po jejich trasách na místo poslední místo doručení objednávky (pokud existuje), kde se bude vozidlo před aktuálním časem nacházet.

## Simulation

Tlačítko pro spuštění (případně zastavení) simulace. Cílem je simulovat dynamické prostředí, kdy objednávky přibývají v čase. Simulace postupně inkrementuje čas (počet tiků) a podle nastavení generuje nové objednávky. Pravidelně pak spouští vybranou optimalizační metodu, která hledá optimální plán pro danou situaci. Výstupem je, stejně jako u jednorázového spuštění, seznam tras, mapa s jejich grafickým přehledem a různé statistiky a ukazatele.

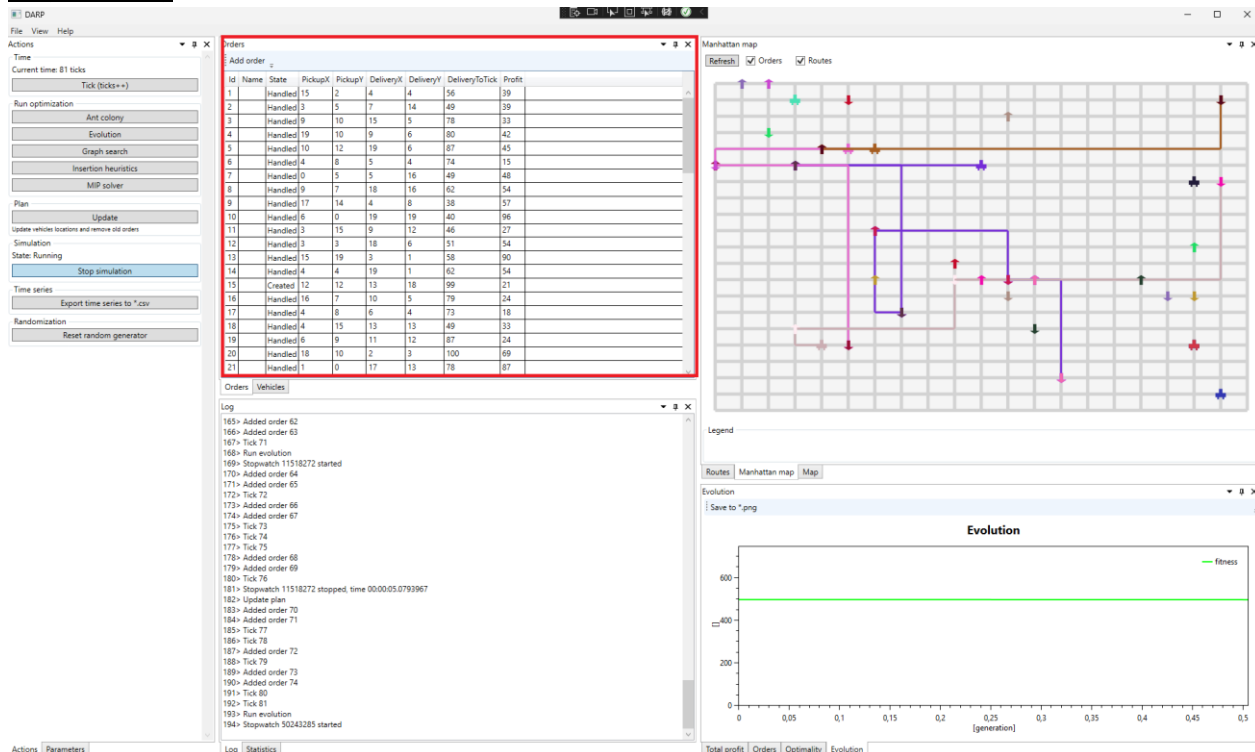
## Time series

Nabízí tlačítko pro export časových řad, které jsou podkladem pro zobrazované grafy v sekci „Statistics“.

## Randomization

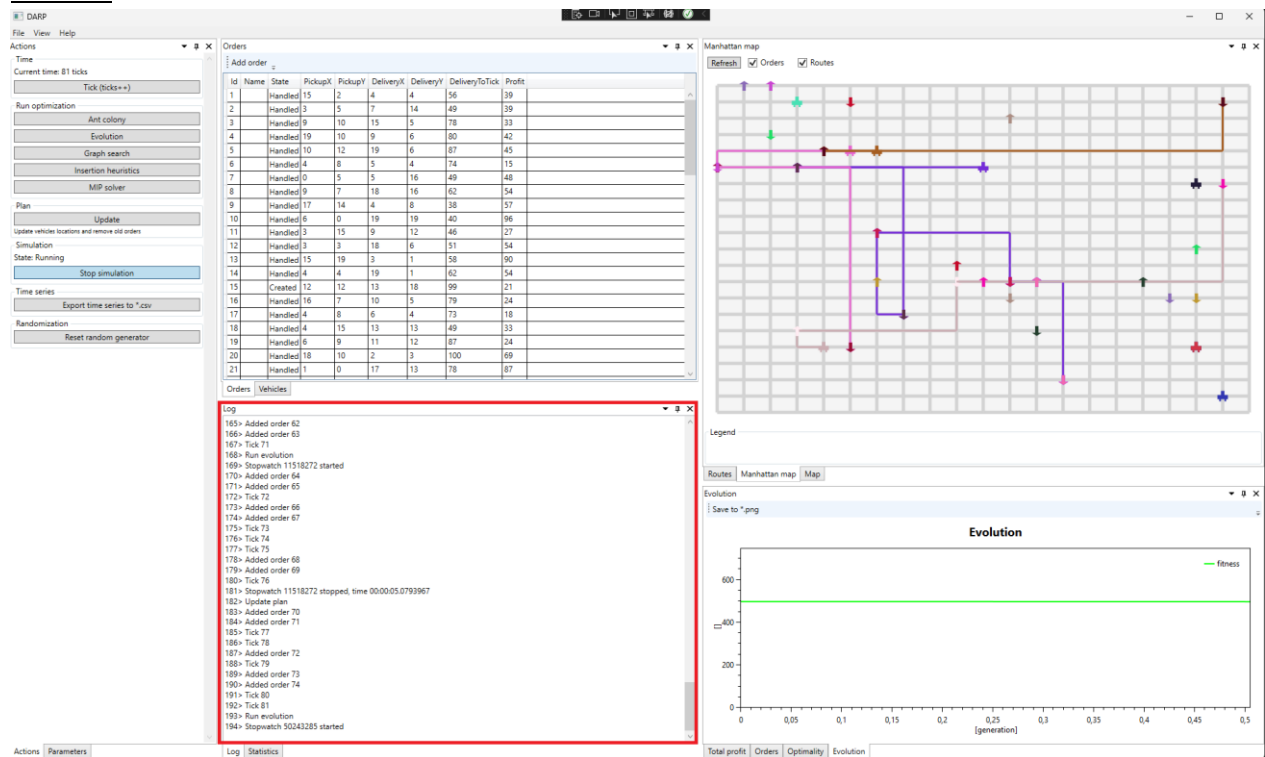
Umožňuje resetovat generátor náhodných čísel se zadaným seedem. Tento generátor je použit pro generování objednávek a při zadání stejného seedu se tedy program bude chovat v tomto ohledu deterministicky. Je však třeba mít na paměti, že některé řešiče jsou nedeterministické, a tedy i pro stejný vstup může program dávat různé výsledky.

## Objednávky



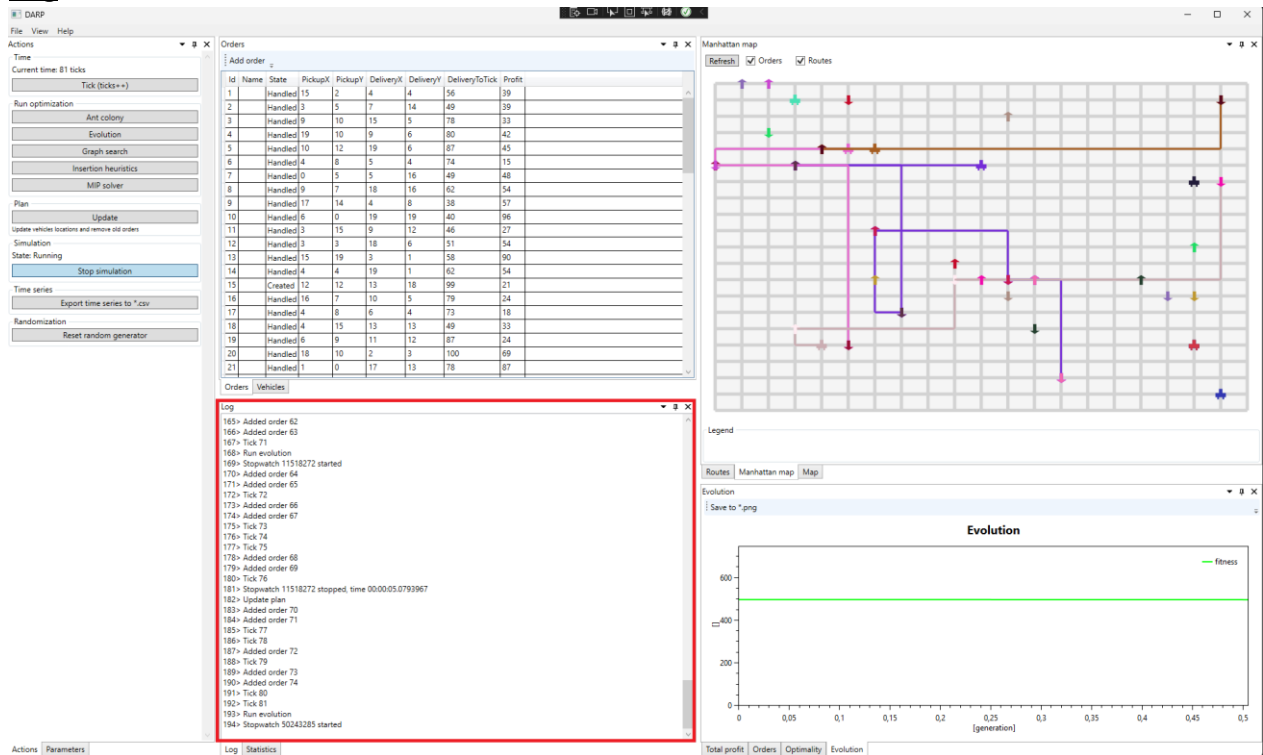
Nové objednávky buď přibývají v průběhu simulace, nebo je lze přidat ručně. Ruční přidání je možné stisknutím tlačítka „Add order“ nebo ručním zapsáním jejích vlastností do seznamu. Id a State jsou obsluhovány automaticky, nelze je zadat ani měnit. Name je volitelné. PickupX/Y a DeliveryX/Y určují souřadnice v mapě, kam má být objednávka převezena. Standartně se pracuje na mřížce s manhattanskou metrikou. DeliveryToTick určuje nejzazší čas, do kdy musí být objednávka doručena, tedy vozidlo ji musí vyzvednout v místě Pickup a převést do místa Delivery. Profit pak udává, kolik je za objednávku zapláceno. Všechny optimalizační metody maximalizují celkový profit.

## Vozidla



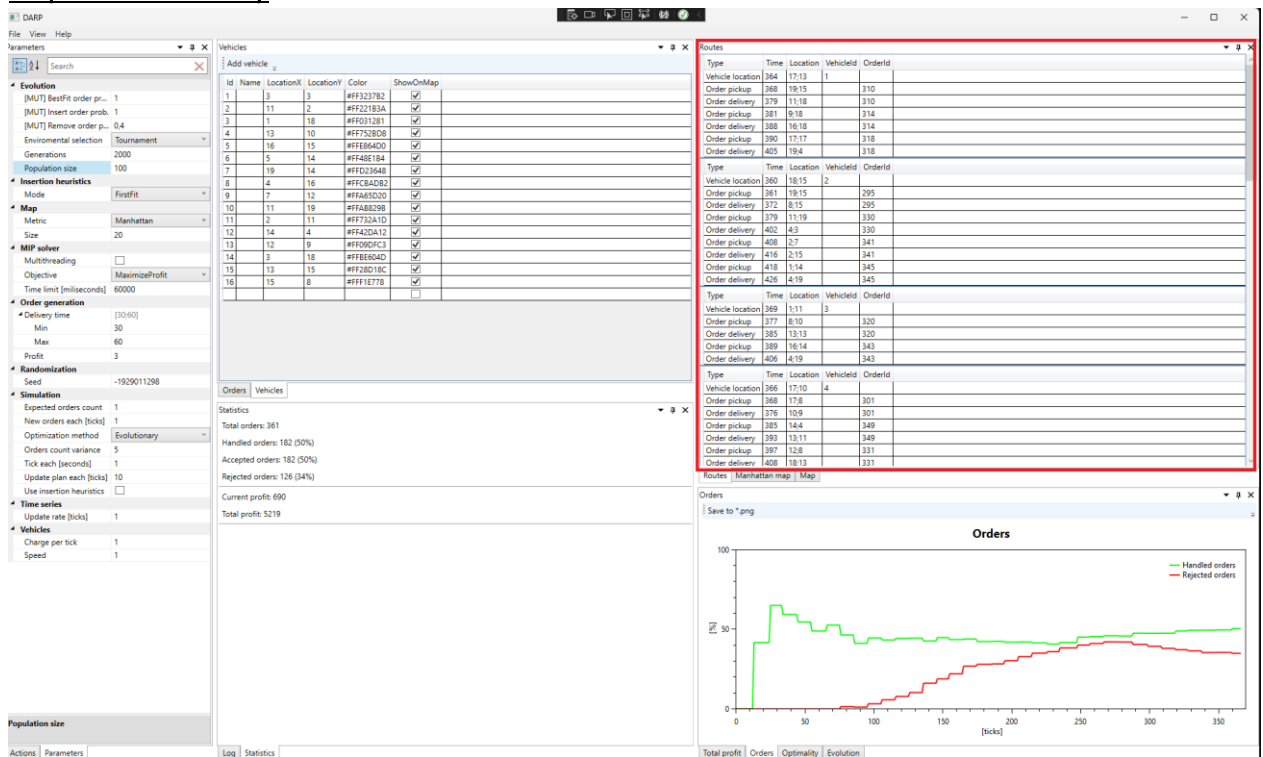
Nová vozidla je nutné přidat před spuštěním simulace a lze je přidávat i v průběhu. Vozidlo má volitelné pole Name a je nutné zadat výchozí pozici. Dále jsou k dispozici parametry Color, který určuje barvu trasy na mapě a ShowOnMap, na základě kterého se vozidlo zobrazí na mapě.

# Log



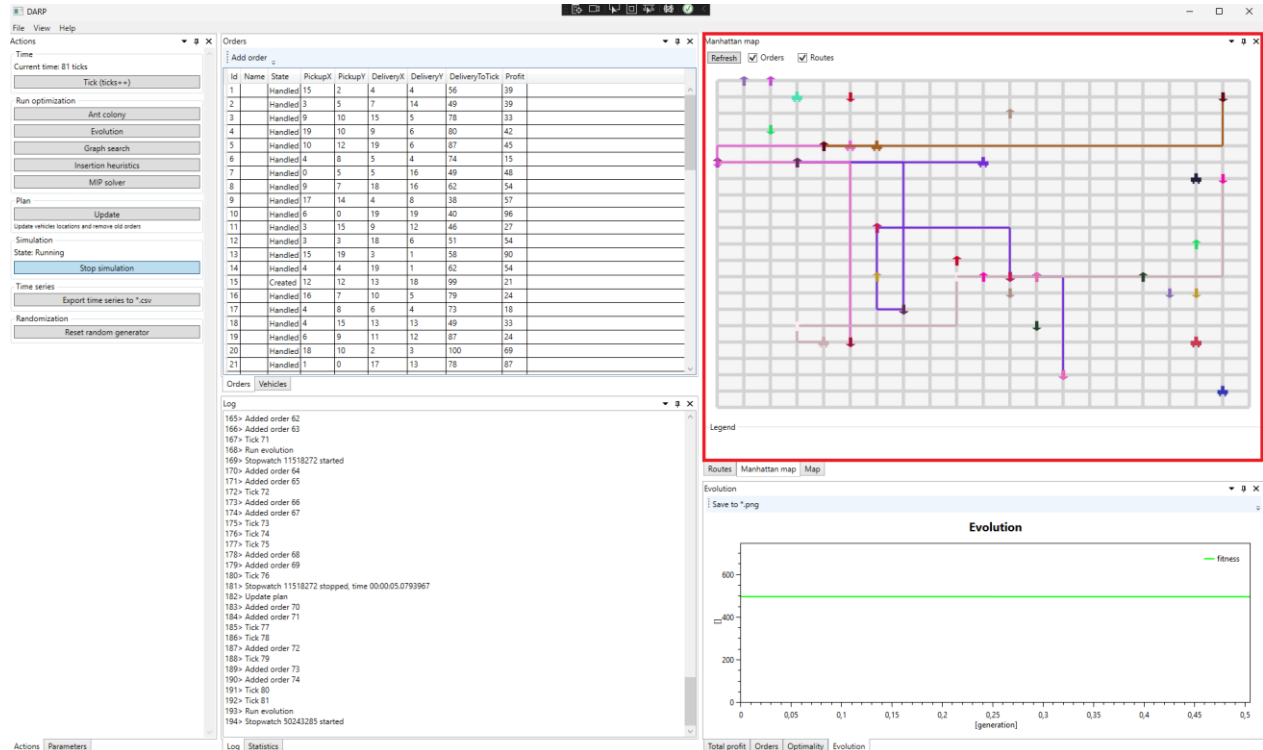
V logu je vidět seznam většiny důležitých akcí, které program provádí a je něm možné online sledovat průběh simulace.

## Naplánované trasy



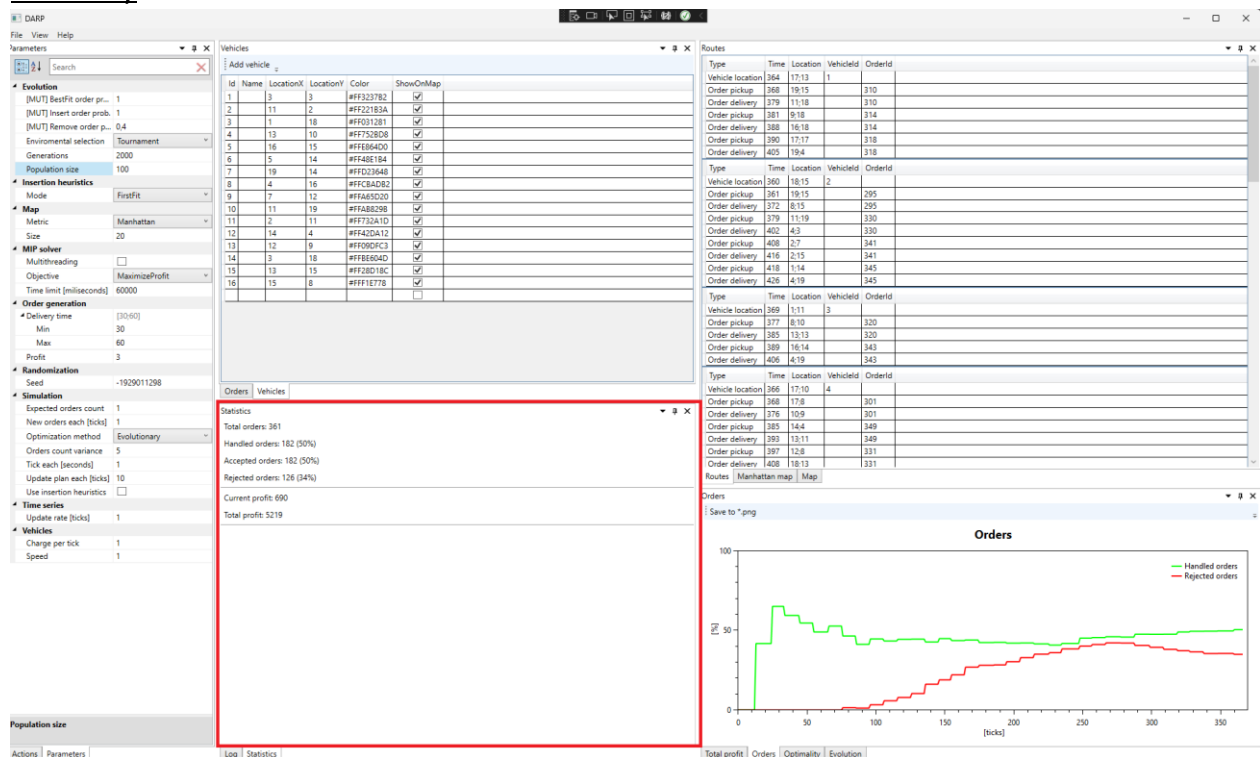
Seznam tras naplánovaných algoritmem. Trasy jsou vypsány pod sebou po jednotlivých vozidlech. Z tras se průběžně odstraňují objednávky, které byly doručeny před aktuálním časem a vozidlo se posouvá do míst jejich doručení. To lze vyvolat i ručně tlačítkem „Update plan“ v sekci akcí.

## Mapa (Manhattan)



Mapa zobrazuje na mřížce s manhattanskou metrikou objednávky, vozidla a jejich trasy. Pickup objednávky je značen ↑, delivery pak ↓. U každého vozidla je možné v seznamu zaškrtnout, zda se má na mapě zobrazovat. Mapa se neaktualizuje automaticky, ale je třeba vyvolat její překreslení tlačítkem „Refresh“.

## Statistiky



V sekci statistik jsou v horní části vidět základní ukazatele, v dolní se pak vykreslují grafy. Aktuálně jsou k dispozici 4 různé grafy. Časové řady, z nichž se grafy kreslí, lze vyexportovat tlačítkem „Export time series to \*.csv“ v sekci akcí.

Grafy:

1. Total profit – celkový zisk v průběhu času. Celkový zisk odpovídá ziskům z doručených (Handled) objednávek minus náklady na jízdu vozidel. Oba tyto parametry lze nastavit v sekci s parametry.
2. Orders – poměr doručených (Handled) a nedoručených (Rejected) objednávek. Lze sledovat, jak optimálně nastavit počet vozidel, aby byl například co nejmenší, ale nebylo příliš mnoho nedoručených objednávek.
3. Optimality – zobrazení indexů optimality pro porovnání jednotlivých optimalizačních metod. Pro profit se počítá jako celkový profit/profit při doručení všech objednávek. Pro čas se pak počítá jako čas nutný pouze pro doručení všech objednávek/celkový čas jízdy všech vozidel.
4. Evolution – na tomto grafu je vidět vývoj fitness v průběhu generací při použití evolučního algoritmu. Graf se aktualizuje zhruba každých 50-100 generací.



# Optimalizační metody

## Insertion heuristics

Základní použitou metodou je insertion heuristika, která zařazuje nově příchozí objednávky do již existujících tras. Z nich nic neodebírání, pouze hledá místo pro novou objednávku takové, aby se stihla doručit a zároveň se stihly doručit i všechny po ní následující.

Jako možné vylepšení by bylo umožnit objednávky i odebírat za předpokladu, že se tím stoupne celkový profit.

Aktuální implementace má tři verze, FirstFit, LocalBestFit a GlobalBestFit.

1. **FirstFit** prochází nové objednávky v pořadí od těch, co se musí doručit nejdříve. Každou zařadí na první možné místo do první možné trasy.
2. **LocalBestFit** prochází nové objednávky v pořadí od těch, co se musí doručit nejdříve. Po každou vyzkouší všechna místa, kam je možné ji zařadit a zvolí to, které nejvíce zvýší celkový profit.
3. **GlobalBestFit** prochází nové objednávky a všechny se pokusí zařadit. Nakonec vybere tu, jejíž zařazení nejvíce zvýší celkový profit a tu skutečně zařadí do plánu. Následně se proces opakuje, opět projde všechny zbývající nové objednávky a zařadí

Výhodou této metody je, že je extrémně rychlá.

## Mixed integer programming

Další možnou metodou je využití lineárního programování. K řešení lineárního programu se použije externí řešič SCIP.

Za zvážení by stálo využití lineární relaxace uvolněním podmínek na celočíselnost s čímž by možná šel dokázat nějaký aproximační poměr.

V aktuální implementaci MIP podporuje dvě různé účelové funkce. Jedna minimalizuje celkovou ujetou vzdálenost s podmínkou na obsluhu všech objednávek. Druhá maximalizuje celkový profit a naopak nevyžaduje obsluhu všech objednávek.

Výhodou je, že tato metoda vždy poskytne optimální řešení. Nevýhodou je délka běhu. Pro problémy s velikostí nad cca. 30 objednávek a 5 vozidel je čas běhu v řádu minut, což není příliš praktické.

## Evoluční metoda

Zatím poslední implementovanou metodou je evoluční algoritmus, který postupně vylepšuje a modifikuje aktuální plán. Pracuje se třemi mutacemi, jejichž pravděpodobnosti lze upravit v sekci parametrů.

1. Odstranění objednávky z trasy
2. Zařazení objednávky do náhodné trasy na náhodnou pozici, pokud lze
3. Zařazení objednávky do náhodné trasy na nejlepší možnou pozici (použití BestFit heuristiky)

Nově zmutovaní jedinci jsou zařazeni do populace, čímž se zvětší. Na konci každé generace je pak buďto turnajovou selekcí nebo elitářsky vybrán fixní počet jedinců do další generace.

Ještě je třeba vymyslet a implementovat vhodnou operaci křížení, minimálně pro vyzkoušení.

Aktuálně se také začíná z uniformní populace, kde jsou všichni jedinci stejní. Lepší by bylo pamatovat si populaci z poslední generace a využít tu.

### Ant colonies

Je rozpracováno.

### Kombinované řešení

Zaškrtnutím parametru „Use insertion heuristics“ v sekci „Simulation“ je možné použít tzv. kombinované řešení, kdy se v první řadě pustí heuristika a až pokud se jí nepodaří zařadit všechny nové objednávky, tak je spuštěna standardní metoda podle výběru.

## Příklady

### Jednorázová optimalizace

1. Nastavím vhodně parametry. Zásadními jsou velikost mapy, časové okno generovaných objednávek či profit.
2. Přidám objednávky, buď ručně, nebo si je nechám vygenerovat náhodně tlačítkem „Add order“.
3. Přidám vozidla.
4. Vyberu si optimalizační metodu, například Evoluční algoritmus.
  - a. Nastavím parametry v sekci „Evolution“ jako počet generací, velikost populace či pravděpodobnosti realizace mutací
  - b. Spustím optimalizaci tlačítkem v sekci akcí, ve skupině „Run optimization“.
5. Mohu si prohlédnout navržené trasy a zobrazit si je na mapě
6. Statistiky nejsou k dispozici, protože pracuji v jednom časovém okamžiku (nejspíše 0)
7. Lze pokračovat dále
  - a. Posunu se v čase tlačítkem „Tick“
  - b. Přidám další objednávky
  - c. Kliknu na „Update plan“, čímž se mi aktualizují pozice vozidel
  - d. Znovu spustím optimalizaci (klidně jinou než původně)

### Simulace

1. Nastavím vhodně parametry. Zásadními jsou velikost mapy, časové okno generovaných objednávek či profit.
2. Nastavím parametry simulace v sekci „Simulation“. Zvolím optimalizační metodu a zaškrtnu, zda chci používat insertion heuristiku.
3. Přidám vozidla. Lze si je ihned zobrazit na mapě po stisku tlačítka Refresh.
4. Spustím simulaci tlačítkem „Start“
5. Průběh simulace lze sledovat v logu
6. Parametry lze měnit i v průběhu simulace, jejich změny se projeví okamžitě

## Testování

Níže jsou výsledky testu na statických datech, tedy v jednom časovém okamžiku. Na vstupu bylo 30 objednávek a 5 vozidel na mapě 20x20. Délka běhu SCIPu byla omezena na 30s, řešení tedy není vždy optimem, ale pouze přípustným.

Je vidět, že bezkonkurenčně nejlépe si vede GlobalBestFit heuristika, v závěsu za ní je evoluční algoritmus.

Data jsou ve formátu [metoda] [profit] [čas běhu]

```
----- Run 0 -----
Evolution 507, time 00:00:01.5002891
Insertion heuristics (first fit) 328, time 00:00:00.0006915
Insertion heuristics (local best fit) 367, time 00:00:00.0001541
Insertion heuristics (global first fit) 541, time 00:00:00.0022377
MIP 518, time 00:00:29.9253646
----- Run 1 -----
Evolution 595, time 00:00:01.6034415
Insertion heuristics (first fit) 347, time 00:00:00.0000909
Insertion heuristics (local best fit) 479, time 00:00:00.0001993
Insertion heuristics (global first fit) 607, time 00:00:00.0020250
MIP 556, time 00:00:29.4009486
----- Run 2 -----
Evolution 539, time 00:00:01.3762559
Insertion heuristics (first fit) 274, time 00:00:00.0000833
Insertion heuristics (local best fit) 420, time 00:00:00.0004876
Insertion heuristics (global first fit) 585, time 00:00:00.0015627
MIP 555, time 00:00:29.6202152
----- Run 3 -----
Evolution 457, time 00:00:01.5635087
Insertion heuristics (first fit) 280, time 00:00:00.0000851
Insertion heuristics (local best fit) 338, time 00:00:00.0001907
Insertion heuristics (global first fit) 496, time 00:00:00.0024944
MIP 555, time 00:00:29.4233080
----- Run 4 -----
Evolution 579, time 00:00:01.4012163
Insertion heuristics (first fit) 339, time 00:00:00.0000801
Insertion heuristics (local best fit) 424, time 00:00:00.0001603
Insertion heuristics (global first fit) 608, time 00:00:00.0018169
MIP 619, time 00:00:29.6079995
----- Run 5 -----
Evolution 533, time 00:00:01.4768335
Insertion heuristics (first fit) 328, time 00:00:00.0000886
Insertion heuristics (local best fit) 407, time 00:00:00.0001788
Insertion heuristics (global first fit) 621, time 00:00:00.0019844
MIP 686, time 00:00:29.5151510
----- Run 6 -----
Evolution 509, time 00:00:01.3323062
Insertion heuristics (first fit) 255, time 00:00:00.0000729
Insertion heuristics (local best fit) 392, time 00:00:00.0001727
Insertion heuristics (global first fit) 554, time 00:00:00.0015841
MIP 572, time 00:00:29.6674351
----- Run 7 -----
Evolution 570, time 00:00:01.7085237
Insertion heuristics (first fit) 357, time 00:00:00.0000897
Insertion heuristics (local best fit) 453, time 00:00:00.0001838
Insertion heuristics (global first fit) 575, time 00:00:00.0022052
MIP 664, time 00:00:29.2794041
----- Run 8 -----
Evolution 574, time 00:00:01.7948949
Insertion heuristics (first fit) 280, time 00:00:00.0000871
Insertion heuristics (local best fit) 459, time 00:00:00.0001999
Insertion heuristics (global first fit) 563, time 00:00:00.0022611
MIP 615, time 00:00:29.2034006
----- Run 9 -----
Evolution 498, time 00:00:01.6660548
Insertion heuristics (first fit) 210, time 00:00:00.0001155
Insertion heuristics (local best fit) 465, time 00:00:00.0002424
Insertion heuristics (global first fit) 561, time 00:00:00.0027354
MIP 580, time 00:00:29.3282140
```