

Algorytm: Symulowane Wyżarzanie

Zespół: Antoni Kraczowski, Jakub Kołton

1. Funkcjonalność aplikacji

a) GUI: graficzne przedstawienie rozwiązania, wyświetlanie przygotowanych wykresów

b) Interfejs:

"""

Class for solving problems with the Simulated Annealing method.

:param SolutionType: type of schedule to be returned, used for validation during runtime

:type SolutionType: Type

:param cost: Objective function that is being optimized, takes schedule as input and returns cost

:type cost: Callable returning real value

:param sol_gen: Function that returns schedule generated randomly from the neighbourhood of the provided schedule,

takes the schedule as input and returns new schedule as output

:type sol_gen: Callable returning SolutionTemplate

:param cool: Function that cools the temperature, takes temperature and iteration as parameters

:type cool: Callable returning non-negative value

:param probability: Probability function, returns probability of transition to different state, takes delta_energy

and iterations as parameters

:type probability: Callable returning real value between 0 and 1

:param init_sol: Starting visited_solution for the algorithm

:type init_sol: SolutionTemplate

:param init_temp: Starting temperature

:type init_temp: Real value

:param max_iterations: Max number of iterations of the main loop

:type max_iterations: int

:param experiment_name: name of the experiment

:type experiment_name: str

:param log_file_path: path of the log file

:type log_file_path: str

:param csv_file_path: path of the csv file

:type csv_file_path: str

:param log_results: flag whether to log results

:type log_results: bool

:param remember_iteration: flag whether to remember the vector of iterations during runtime

:type remember_iteration: bool

:param remember_temperature: flag whether to remember vector of temperature during runtime

:type remember_temperature: bool

:param remember_delta_energy: flag whether to remember vector of delta_energy during runtime

:type remember_delta_energy: bool

:param remember_probability_of_transition: flag whether to remember vector of probability during runtime

:type remember_probability_of_transition: bool

:param remember_cost_function: flag whether to remember vector of cost function during runtime

:type remember_cost_function: bool

:param remember_best_cost_function: flag whether to remember vector of best cost function during runtime

:type remember_best_cost_function: bool

:param remember_visited_solution: flag whether to remember vector of visited schedule during runtime

:type remember_visited_solution: bool

""

c) Narzędzia zewnętrzne: Excel/MatLab

2. Prezentacja własności rozwiązania

- a)** Graficzna reprezentacja przypisania pracowników do maszyn w danych jednostkach czasowych(GUI)
- b)** Wykresy wartości parametrów algorytmu zebrane w trakcie jego przebiegu

3. Prezentacja własności algorytmu

- a)** Przebieg wartości parametrów w trakcie symulacji(temperatura, funkcja celu)
- b)** Mechanizmy: mechanizmy pomocnicze przy generowaniu sąsiadów, importer danych, eksporter wyników do formatu csv, plik z testami jednostkowymi dla wszystkich klas w projekcie
- c)** Elementy algorytmu(u nas klasa Solver):
 - walidatory atrybutów przesyłanych do algorytmu
 - metody ustawiające ścieżki do plików logujących(logi, wyniki)

4. Eksperymenty obliczeniowe

- a)** Instancje testowe: format .json, zgodny z interfejsem ResourceManager, generowane modelem językowym. 10 maszyn 20 pracowników 7 dni, dane rozłożone równomiernie, przypadek graniczny: duże zapotrzebowanie na maszynie z ograniczonymi maksymalnymi pracownikami
- b)** Metodyka badań: badanie zachowania się przebiegu wartości parametrów symulacji dla wielu wywołań algorytmu(100/1000) i stałych danych wejściowych
- c)** Problemy badawcze: wpływ temperatury początkowej na zachowanie się algorytmu, wpływ wyboru schematu chłodzenia, wpływ różnych typów sąsiedztw. Sposób badania zawsze taki sam, czyli puszczenie algorytmu wiele razy i sprawdzenie się zachowania.
- d)** Zapamiętywane będą: temperatura, funkcja celu, aktualnie znana najlepsza funkcja celu oraz prawdopodobieństwo przejścia do gorszego rozwiązania.
- e)** Na wykresach będziemy mieć zależności wszystkich zapamiętywanych wartości parametrów algorytmu w kolejnych iteracjach. Wykresy badania zachowania się algorytmu będą sprawdzały jak średnio algorytm poprawia nasze rozwiązanie w zależności od przyjętych elementów w danej instancji testowej.