

## O projekcie

Na zlecenie firmy P2ware zespół zaprojektował i zaimplementował bibliotekę do optymalizacji planów projektów. P2ware Project Planner Suite jest komercyjnym oprogramowaniem o szerokich zastosowaniach w zakresie planowania projektów, sterowania realizacją, monitorowania, raportowania i kontroli. Wykonaliśmy analizę problemu, w celu uwzględnienia możliwie wielu elementów planowanych struktur, ograniczeń i preferencji użytkownika oraz możliwości obliczeniowych. Zaimplementowana biblioteka dostarcza kilku algorytmów, które w ramach określonych przez użytkownika znajdują plan projektu optymalny pod względem czasu zakończenia, trwania, spójności i kosztów.

## Opis problemu

Każdemu **działaniu** należy przydzielić odpowiednie **zasoby** i ustalić kiedy ma być wykonywane. Danymi wejściowymi systemu są m.in.:

- ▶ **Działania** – zadania do wykonania, np. *zaprogramowanie modułu*. Działanie ma określony z góry łączny czas trwania, kalendarz, podczas którego może być wykonywane, dodatkowe ograniczenia czasowe, wymagania zasobowe, ...
- ▶ **Zasoby** – **materiał** lub **praca**, np.  *płyty* lub *programiści firmy Foo*. Zasób określa tagi (np. *ekspert baz danych*), kalendarz dostępności, dostępną ilość, koszt aktywacji, koszt za 1h pracy jednostki.
- ▶ **Wymagania zasobowe** – rodzaj, zestaw tagów, ilość.
- ▶ **Diagram następstw** – graf między krańcami działaniami, krawędzie określają wymagane opóźnienie, np. *wdrożenie można zacząć dopiero po zakończeniu instalacji serwerów, testowanie musi zakończyć się tydzień po końcu implementacji (ale może częściowo się pokrywać)*.
- ▶ **Kalendarze** – np. *w maju pon-pt 8:15-12:00 i 14:00-18:00, w czerwcu piątki tylko do 15:00, oprócz majówki, z jeszcze innymi godzinami*.

## Algorytm zachłanny

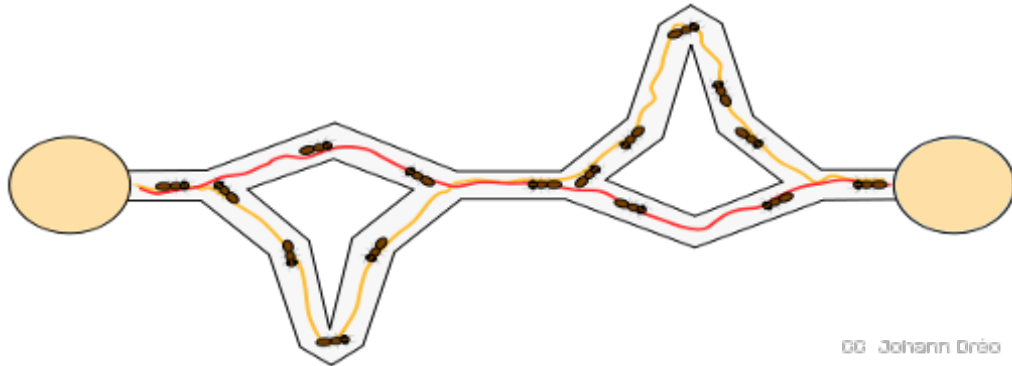
[Greedy]

Dowodzimy, że sam problem istnienia lub aproksymacji rozwiązania jest nawet w najprostszymi wariantach NP-trudny. Przy danej podpowiedzi określającej priorytety działań potrafimy jednak szybko znaleźć pełen plan o optymalnym czasie zakończenia. W tym celu sortujemy topologicznie krańce działań zgodnie z diagramem następstw i priorytetami. Następnie wykonywane są cztery przejścia analizujące w tej kolejności działania i możliwe przydziały zasobów. Algorytm zachłanny bierze na siebie całą skomplikowaną strukturę problemu i konstrukcję pełnego rozwiązania. Pozostaje znaleźć odpowiednie priorytety, czym mogą zająć się pozostałe algorytmy.

## Algorytm mrówkowy

[AntColony]

Inspirowany zachowaniem kolonii mrówek, algorytm szuka priorytetów jako ścieżki. Są one równoległe oceniane, na najlepszych odkładany jest zaś feromon, który jednak z czasem paruje. Początkowo mrówki przeszukują losowe priorytety, z czasem wybierane ścieżki zbiegają do optymalnej.



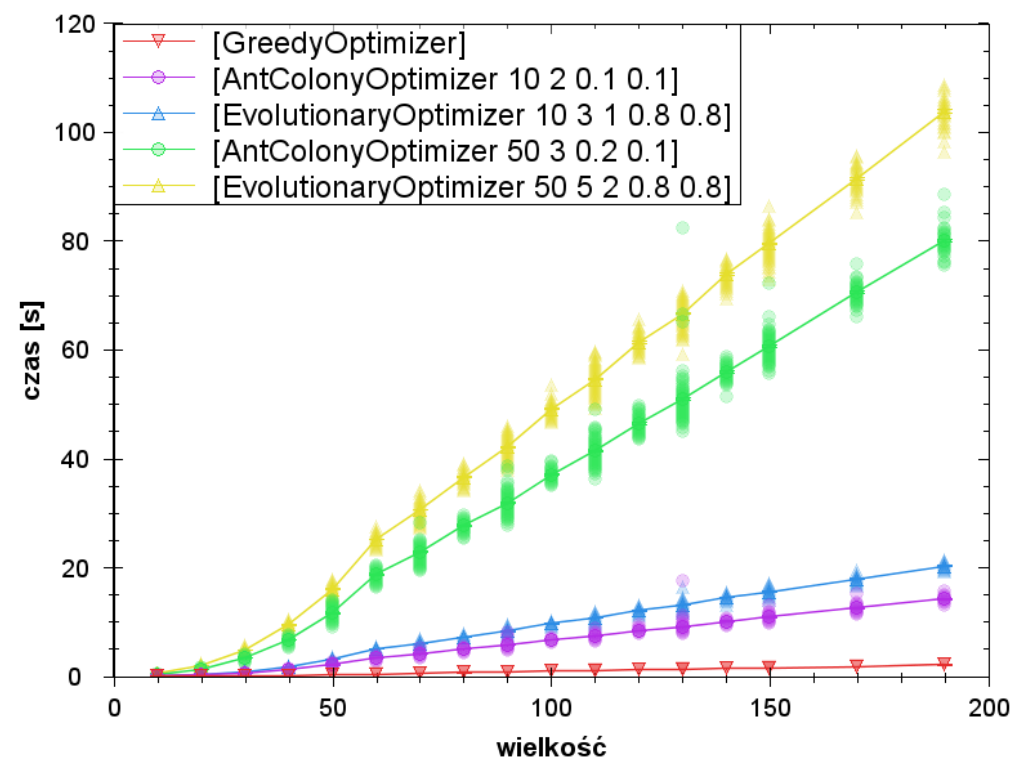
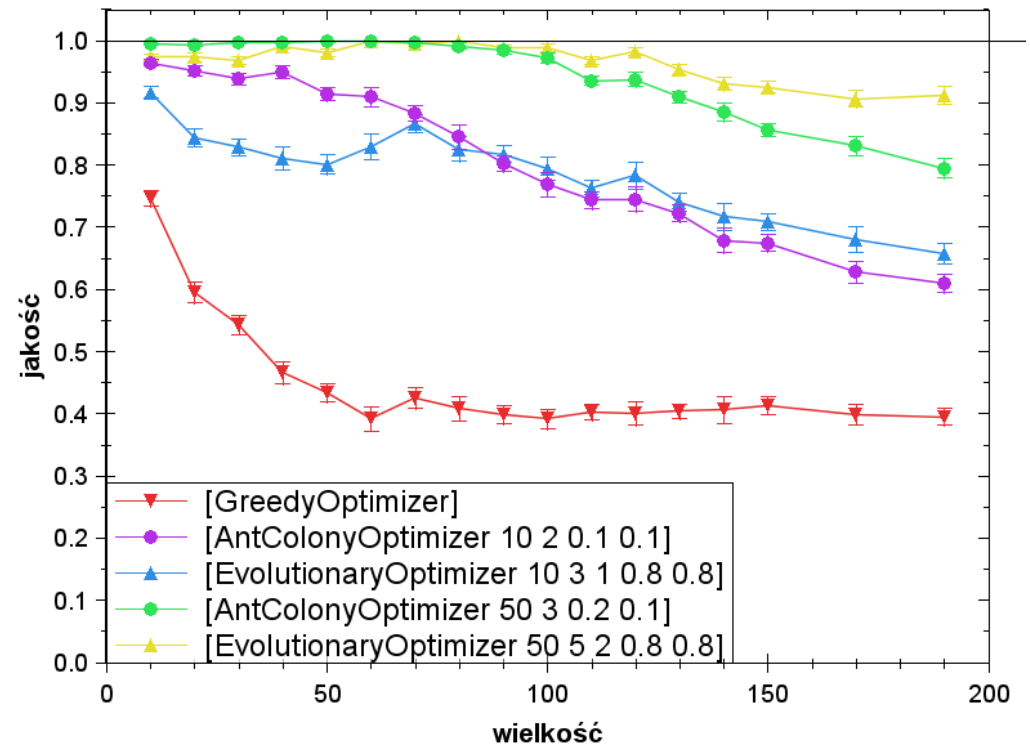
## Algorytm ewolucyjny

[Evolutionary]

Nasz najlepszy algorytm opiera się na metaheurystyce ewolucyjnej. Podpowiadane kolejności działań są cyklicznie klonowane i mutowane o wielkość malejącą z czasem. Obliczenie jakości jednej puli następuje równoległe, po czym z  $\alpha$  najlepszych tworzona jest nowa pula.

## Wyniki

Testy generujemy m.in. przez redukcję problemu MAXCLIQUE, jakość oznacza stosunek rozmiaru znalezionej kliki do rozmiaru maksymalnej. Wielkość to liczba wierzchołków, czyli działań, krawędzie odpowiadają zasobom i wymaganiom (jest ich zawsze 30 razy więcej). Przy nazwach podane są parametry, pierwszy to liczba iteracji. Poniżej jakość uśredniona na 250 testach:



Poszczególne kropki pokazują wyniki pojedynczych testów. Poniżej różne optymalizatory na testach o 300 działaniach i 9000 wymagań:

