OPTYMALIZACJA WYPEŁNIENIA PRZESTRZENI ŁADUNKOWEJ Z WYKORZYSTANIEM ALGORYTMÓW MRÓWKOWYCH

OPTIMIZATION OF LOADING FACILITY WITH THE USE OF ANT COLONY OPTIMIZATION ALGORITHMS

Zbigniew ŚWIĄTNICKI
z.swiatnicki@wsosp.pl

Lotnicza Akademia Wojskowa
Wydział Bezpieczeństwa Narodowego i Logistyki
Instytut Logistyki i Organizacji Transportu

STRESZCZENIE

Artykuł poświęcony jest problemowi wykorzystania przestrzeni ładunkowej. Zastosowano do tego celu algorytmy mrówkowe. Zbudowano model optymalizacyjny oparty na problemie plecakowym. Wybrano 8 różnych algorytmów mrówkowych. Przedstawiono i omówiono uzyskane wyniki zastosowania algorytmów mrówkowych do rozwiązywania problemu plecakowego.

SUMMARY

The article is devoted to the optimization problem of cargo space. Formal algorithms have been used for this purpose. An optimization model based on a knapsack problem was built. 8 different ant algorithms were selected. The results of the application of ant algorithms for solving the knapsack problem are presented and discussed.

Słowa kluczowe: optymalizacja, problem plecakowy, wykorzystanie przestrzeni, algorytm mrówkowy, logistyka
Key words: optimization, knapsack problem, loading facility, ant system, logistics

WSTĘP

Prawidłowe wykorzystanie przestrzeni ładunkowej stanowi przedmiot zainteresowania praktyków logistyków. Ma znaczenie zarówno w magazynowaniu, jak i przemieszczaniu (transporcie) dóbr.

Nauka pośpieszyła z pomocą w rozwiązywaniu tego mającego długą historię zagadnienia. Jednym z najbardziej znanych modeli formalnych jest problem plecakowy. Problem plecakowy jest jednym z najstarszych problemów optymalizacyjnych NP.-zupełnych. Nie odkryto do tej pory metody znajdowania optymalnego rozwiązania tego problemu. Mimo poszukiwań, do dyspozycji pozostają jedynie metody przybliżone, na przykład metody probabilistyczne.

Wbrew pozorom, problem plecakowy jest istotnym zagadnieniem optymalizacyjnym w praktyce logistyki. Spotykamy go w magazynowaniu i transporcie, w tym w transporcie morskim.

W wielu przypadkach rozwiązanie problemu optymalizacji wykorzystania przestrzeni ładunkowej odbywa się na zasadach instrukcyjnych, opracowanych, między innymi, w oparciu o doświadczenie, a także wcześniejsze rozwiązania, niekonieczne optymalne.

Nic stoi jednak nic na przeszkodzie, aby problem optymalizacji wykorzystania przestrzeni ładunkowej, był rozwiązywany w każdym indywidualnym przypadku, w czasie rzeczywistym, a otrzymane rozwiązania były suboptymalnymi (racjonalnymi).

Zagadnienie plecakowe można werbalnie opisać w sposób następujący. Mamy do dyspozycji plecak o maksymalnej pojemności B oraz zbiór N elementów, przy czym każdy element ma określoną wartość oraz wielkość. Problem polega na znalezieniu takiego upakowanego plecaka, aby suma wartości znajdujących się w nim elementów była jak największa. Problem plecakowy często przedstawia się jako problem złodzieja rabującego sklep – znalazł on N towarów, j-ty przedmiot jest wart c_j oraz waży w_j. Złodziej dąży do zabrania ze sobą jak najwartościowszego łupu, przy czym nie może zabrać więcej niż B kilogramów. Nie może też zabierać ułamkowej części przedmiotów.

Jednym z narzędzi umożliwiających rozwiązanie problemu plecakowego są algorytmy mrówkowe. Jest to zagadnienie oryginalne w obszarze systemów mrówkowych.. Większość problemów, do rozwiązywania których wykorzystywane są algorytmy mrówkowe (na przykład klasyczny problem komiwojażera), oparte jest o wyszukiwanie drogi w grafie, drogi zbudowanej z jego krawędzi (na krawędziach pozostawiany jest feromon). W problemie plecakowym kolejność zabieranych przedmiotów nie ma znaczenia – zatem droga przeniesiona musi być na wierzchołki i to na wierzchołkach grafu odkładany jest feromon. Wymaga to zupełnie innego podejścia do problemu.

1. PROPONOWANA METODA OPTYMALIZACJI WYKORZYSTANIA PRZESTRZENI ŁADUNKOWEJ

Obserwacje zachowań stadnych owadów zostały przeniesione w wymiar abstrakcyjny – zbudowano modele symulacyjne systemów mrówkowych. W 1992 roku Marco Dorigo w swojej pracy doktorskiej opisał system mrówkowy:

Wirtualne (symulowane) mrówki różnią się od naturalnego pierwowzoru tym, że:

- Czas w świecie mrówek wirtualnych nie jest ciągły, a dyskretny.
- Mrówki posiadają pamięć, w której zapamiętują odwiedzone przez siebie wierzchołki.
- Mrówki sztuczne posiadają „wzrok” umożliwiający im określenie odległości do najbliższego wierzchołka.
- Feromon w świecie mrówek wirtualnych nie musi być rozkładany ciągle, a w rozmaity sposób, np. dopiero po znalezieniu pełnego rozwiązania – w zależności od zastosowanego algorytmu i jego implementacji.

Odnajdowanie drogi przez algorytm mrówkowy można w najprostszy sposób przedstawić następująco:

- Wybierz (w taki sam sposób dla wszystkich mrówek) wierzchołek początkowy.
- Użyj feromonu i wartości heurystycznych do zbudowania drogi poprzez dodawanie kolejnych wierzchołków do drogi, zgodnie z naturą rozwiązywanego problemu.
- Po znalezieniu drogi rozłóż feromon na wierzchołkach zgodnie z regułami danego algorytmu.
- Powtarzaj czynności aż do spełnienia kryterium stopu[1].

Większość algorytmów mrówkowych stanowią modyfikacje pierwszego z nich – *Ant System* Marco Dorigo. Podstawowa różnica między algorytmami to sposób, w jaki określana jest ilość i miejsce rozkładania feromonu.

Kluczem do sukcesu jest komunikacja między mrówkami. Podczas marszu z mrowiska do źródeł pożywienia i z powrotem, mrówki rozkładają na ziemi substancje zapachową zwaną *feromonem*, tworząc tym samym *ścieżki feromonowe* (Boryczka, 2006), (Dorigo i Stutzle, 2004), (Jankowiak, 2003) 10]. Mrówki potrafią wyczuć feromon i mają tendencję do wybierania z większym prawdopodobieństwem drogi oznaczonej większą ilością feromonu. Jest to rodzaj komunikacji pośredniej zwanej inaczej *stygmergią* (Dorigo i Stutzle, 2004).

1.1. Wykorzystane algorytmy mrówkowe

Do porównań zostały wytypowane następujące algorytmy mrówkowe:

- Density Ant System (DAS);
- Quantity Ant System (QAS);
- Cycle Ant System (CAS);

[1] Na przykład, upływ przewidzianego czasu obliczeń, wykonania ustalonej liczby iteracji, uzyskanie określonej poprawy rozwiązania.

- Elitist Ant System (EAS);
- Max-Min Ant System (MMAS);
- Rank-Based Ant System (AS_{rank});
- Ant Colony System (ACS);
- Ant-Q (AQ).

Powyższe algorytmy zastosowano do rozwiązywania zdefiniowanego problemu optymalizacji wykorzystania przestrzeni ładunkowej.

1.2. Wartość heurystyczna i konstruowanie rozwiązań

Ponieważ w problemie plecakowym nie ma znaczenia kolejność odwiedzanych wierzchołków, to – w odróżnieniu od innych problemów optymalizacyjnych - wartości feromonowe przeniesione są z krawędzi na wierzchołki. Również z wierzchołkami będzie związana wartość heurystyczna. W tym wypadku będzie ona wynosiła:

$$\eta_j = {}^{c_j}/_{w_j},$$

gdzie:

- c_j – oznacza wartość (cenność) przedmiotu j-tego;
- w_j – oznacza wagę (koszt uzyskania/objętość) przedmiotu j-tego.

Atrakcyjność drogi w grafie poszukiwań jest wprost proporcjonalna do wartości zabranego przedmiotu, a nie - jak w większości zastosowań algorytmów mrówkowych – odwrotnie proporcjonalna do długości odnalezionej drogi. Zmodyfikowany jest więc sposób odkładania (przyrostu) feromonu na wierzchołkach podczas rozwiązywania tego problemu. Odpowiednio, dla AC będzie to:

$$\Delta\,\tau_i^k = \begin{cases} Q \cdot C_k & \textit{jeśli przedmiot i został zabrany} \\ 0, & \textit{w przeciwnym wypadku} \end{cases},$$

gdzie C_k oznacza całkowity koszt przedmiotów zabranych do plecaka. W metodzie rankingowej otrzymamy:

$$\Delta\tau_{ij}^r = e \cdot C^r,$$

A z kolei we wszystkich metodach odkładających feromon na najlepszym rozwiązaniu:

$$\Delta\tau_{ij}^{bs} = e \cdot C^{bs}.$$

2. PORÓWNANIE SKUTECZNOŚCI ALGORYTMÓW

Zasadniczym pytaniem jest czy bardziej skuteczny jest algorytm, który znajduje jakościowo lepsze rozwiązanie, czy ten, który odnajduje rozwiązanie gorsze, ale za to w krótszym czasie?

Należy uznać, że istotne są oba aspekty – zarówno jakość uzyskanego rozwiązania, jak i czas, który był potrzebny, aby rozwiązanie uzyskać. Najlepszym algorytmem byłby zatem taki, który uzyskiwałby idealne rozwiązanie w błyskawicznym czasie. W rzeczywistości mamy prawie zawsze do czynienia z koniecznością znalezienia kompromisu.

Miarą skuteczności algorytmów mogą być:

– *jakość znalezionego rozwiązania* – czyli wartość najlepszego, znalezionego przez algorytm, rozwiązania przybliżonego. Miara ta jest stosowana w zasadzie zawsze, gdy mowa jest o ocenie skuteczności algorytmu;
– *koszt znalezienia najlepszego rozwiązania* - czyli liczba iteracji, bądź czas procesora potrzebny do uzyskania najlepszego rozwiązania.

Wielu autorów (Jankowiak, 2003), (Linda i Abrich, 2001) sądzi, że zastosowanie tych dwóch miar nie jest wystarczające do oceny skuteczności algorytmów.

Ponieważ zdarzają się sytuacje, gdy potrzebne jest rozwiązanie o zadanej jakości, bądź dysponujemy ograniczonym czasem, dwie powyższe miary są często poszerzane o dodatkowe:

– *koszt znalezienia rozwiązania o zadanej jakości* – liczba iteracji, bądź czas procesora, jakie są potrzebne do uzyskania rozwiązania o z góry zadanej jakości. W sytuacji gdy algorytm nie jest w stanie uzyskać rozwiązania o zadanej jakości, wartość tej miary przyjmuje się jako nieskończoność.
– *jakość uzyskanego rozwiązania po zadanej liczbie iteracji* – czyli ocena najlepszego znalezionego rozwiązania, po wykonaniu przez algorytm zadanej liczby iteracji (upływie zadanego czasu).

W przypadku algorytmów populacyjnych, jakimi są algorytmy mrówkowe korzysta się z miary nazywanej średnim rozwiązaniem.

*Średnie rozwiązanie***,** czyli średnia arytmetyczna ocen wszystkich odnalezionych rozwiązań. W tym przypadku miara ta jest bardziej wiarogodna niż miara w postaci najlepszego uzyskanego rozwiązania, gdyż algorytmy często zupełnie heurystycznie (ale nie przypadkowo) mogą znaleźć bardzo dobre rozwiązanie, natomiast większość poszukiwań odbywa się w obszarach, gdzie rozwiązania są dalekie od optymalnych. Średnie rozwiązanie to:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

gdzie:

n – liczba pomiarów

x_i – i-ty wynik pomiaru

Powyższe proste miary umożliwiają ocenę skuteczności algorytmów mrówkowych. Jednak nie dają obrazu postępu poszukiwania rozwiązań. A może mieć to olbrzymie znaczenie dla sterowania realizacją algorytmów. Ważne są zachowania algorytmu w fazie inicjacji, w etapie iteracyjnego poszukiwania rozwiązań i dojście do wyniku końcowego, a nie tylko wynik ostateczny.

W niektórych przypadkach wynik zależy od decyzji w początkowej fazie iteracji. Czasami ostateczne wyniki, są ustalane już w pierwszej iteracji, a w innych przypadkach algorytm potrafi przeszukiwać przestrzeń rozwiązań bardzo długo. Dlatego jakość rozwiązania i czas jego uzyskania to za mało aby ocenić skuteczność algorytmu. Niektóre algorytmy mogą odnajdować rozwiązania szybko, ale w zasadzie z przypadkową dokładnością (różną w kolejnych próbach), inne mogą szukać dłużej, ale znalezione rozwiązania mogą być zawsze podobnej jakości (wyższa stabilność).

Aby ocenić algorytmów pod kątem stabilności rozwiązań lub zdolności do unikania ekstremów lokalnych, można wykorzystać dwa pojęcia:

– *intensyfikacja* (inaczej eksploatacja), czyli zdolność algorytmu do efektywnego badania obszaru poszukiwań i znajdowania rozwiązań o wysokiej jakości w niewielkim obszarze poszukiwań;
– *dywersyfikacja* (inaczej eksploracja), czyli zdolność algorytmu do przechodzenia do niezbadanych obszarów poszukiwań (opuszczanie obszarów ekstremów lokalnych).

Można posłużyć się miarami zaproponowanymi w (Mills, 2000):

– *liczba kroków pomiędzy lokalnymi minimami* – określa ilość czasu niezbędną do przejścia od jednego suboptymalnego rozwiązania do innego. Często algorytmy heurystyczne zatrzymują się w ekstremum lokalnym, ale potrafią je opuścić w poszukiwaniu ekstremum globalnego. Miara ta określa liczbę kroków algorytm niezbędną do takiego przejścia. Im liczba mniejsza, tym dywersyfikacja wyższa;

- *liczba powtórzeń podczas poszukiwania* – określa czy algorytm ma pętle – czy wraca do odwiedzonych już przez obszarów przestrzeni poszukiwań. Wysoka wartość tego wskaźnika oznacza, że dywersyfikacja jest zbyt wysoka. Z kolei niska wartość oznacza zbyt wysoką intensyfikację;
- *odchylenie standardowe najlepszego znalezionego rozwiązania*, to najbardziej klasyczna miara zmienności. Jest to miara określająca, jak bardzo wartości rozwiązania są skoncentrowane wokół rozwiązania średniego. Im mniejsza wartość tym rozwiązania są zbliżone do średniej (skupione).

Dla skończonej populacji przyjmuje postać:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i-\bar{x})^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n}x_i^2}{n} - \bar{x}^2},$$

gdzie:

x_i - oznacza kolejne wartości danej zmiennej,

$\bar{x}$ - to średnia arytmetyczna wartości zmiennej,

n – liczba elementów w populacji rozwiązań.

Oczywiście, im większe odchylenie standardowe tym większa dywersyfikacja. Odchylenie standardowe można interpretować jako miarę niepewności pomiarów. W momencie, gdy osiągnie wartość 0 – wszystkie pomiary są takie same.

Entropia, zwana miarą nieokreśloności. Entropię obliczamy jako:

$$H(x) = \sum_{i=1}^{p} p(i) log_r \frac{1}{p(i)} = -\sum_{i=1}^{n} p(i) log_r p(i),$$

gdzie:

$p(i)$ - prawdopodobieństwo zajścia zdarzenia i.

Entropia określa jak dokładnie rozpatrywane rozwiązania pokrywają przestrzeń rozwiązań. Przyjmuje wartości z przedziału [0, 1]. Jest maksymalna, gdy prawdopodobieństwo zdarzeń jest równe dla wszystkich zdarzeń. Przyjmuje wartość 0, gdy prawdopodobieństwo zajścia jakiegoś zdarzenia wynosi 1.

Im wyższa wartość entropii, tym dywersyfikacja algorytmu jest wyższa: dla wartości 1 rozwiązania pokrywają równomiernie całą przestrzeń poszukiwań, z kolei dla entropii równej 0 – sprowadzają się do jednego rozwiązania.

Ocena wyspecyfikowanych w punkcie 1.1 algorytmów została dokonana w dwóch omówionych powyżej dwóch aspektach.

Skuteczność algorytmów

Skuteczność algorytmów, mierzona dwoma parametrami:

- *jakość najlepszego znalezionego rozwiązania;*
- *czasem potrzebnym na znalezienie najlepszego rozwiązania*.

Eksploracja algorytmów

Sposób przeszukiwania przestrzeni rozwiązań, określany na podstawie:

- *średniej jakości znalezionego rozwiązania;*
- *odchylenia standardowego najlepszych rozwiązań;*
- *entropii.*

3. PORÓWNANIE SKUTECZNOŚCI ALGORYTMÓW MRÓWKOWYCH ZASTOSOWANYCH DO OPTYMALIZACJI WYKORZYSTANIA PRZESTRZENI ŁADUNKOWEJ

3.1. Jakość uzyskanego rozwiązania

Analizując wyniki uzyskane przy użyciu różnych algorytmów mrówkowych, można zauważyć, że żaden z nich nie jest dominujący. Najlepsze rozwiązanie uzyskał algorytm MMAS, jednak różnica między nim i pozostałymi algorytmami jest niewielka.

Dużo gorsze wyniki zostały uzyskane przy użyciu algorytmów DAS i QAS. Nie należy spodziewać się, że któryś z tych algorytmów uzyska lepsze wyniki w miarę jego doskonalenia. Skupić więc należy się nad pozostałymi algorytmami.

Tabela 1. Porównanie jakości rozwiązań uzyskanych przez poszczególne algorytmy

Lp.	Nazwa algorytmu	Jakość uzyskanego rozwiązania
1	Max-Min Ant System	2994
2	Ant-Q System	2992
3	Rank-Based System	2992
4	Ant Colony System	2991
5	Cycle Ant System	2990
6	Elitist Ant System	2990
7	Density Ant System	2818
8	Quantity Ant System	2773

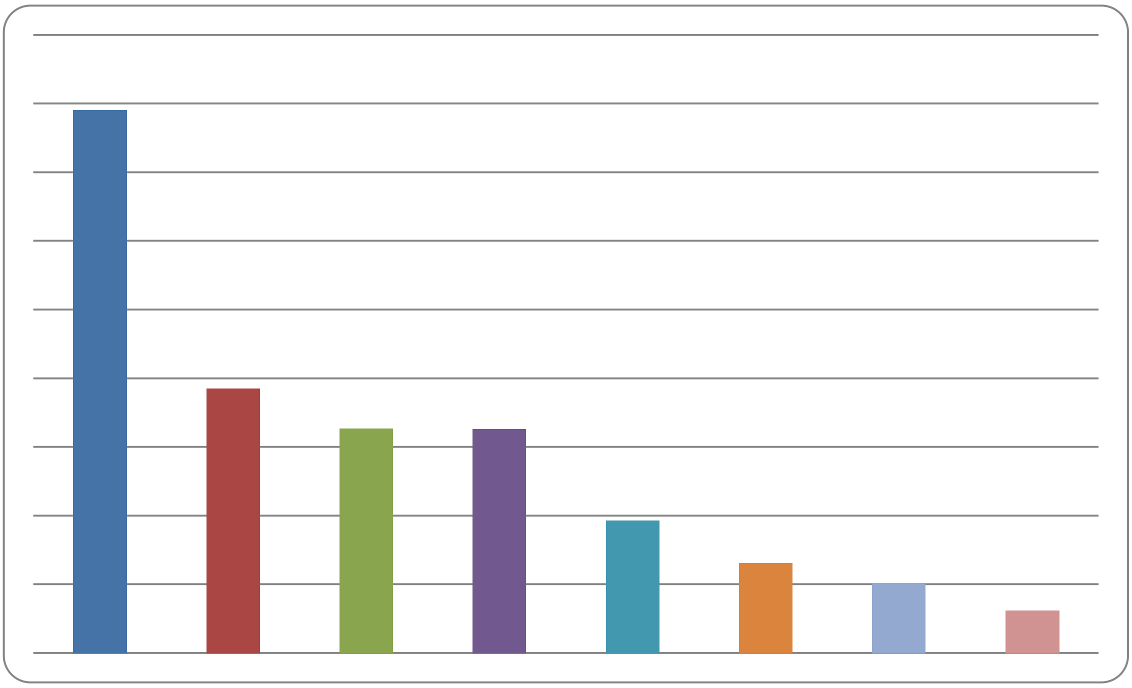
Źródło: Krajewski, 2007.

3.2. Czas uzyskania najlepszego rozwiązania

Najwięcej czasu na wyszukiwanie rozwiązania potrzebuje algorytm MMAS. Uzasadnione jest to tym, że musi on wykonać 200 iteracji, podczas gdy inne algorytmy mogą zatrzymać się wcześniej Algorytm EAS znajduje rozwiązanie w czasie o połowę krótszym.

Najszybciej znajdującym rozwiązanie jest algorytm ACS. Uzasadnione jest to tym, że zarówno odkładanie, jak i parowanie feromonu odbywa się tylko na najlepszej znalezionej do tej pory ścieżce.

Warto zauważyć, że algorytm AS_{rank} — trzeci z najlepszych - na znalezienie rozwiązania problemu potrzebuje średnio zaledwie około 2 sekund. Uzyskuje wyniki podobnie dobre jak algorytm EAS, który potrzebuje aż czterokrotnie więcej czasu.

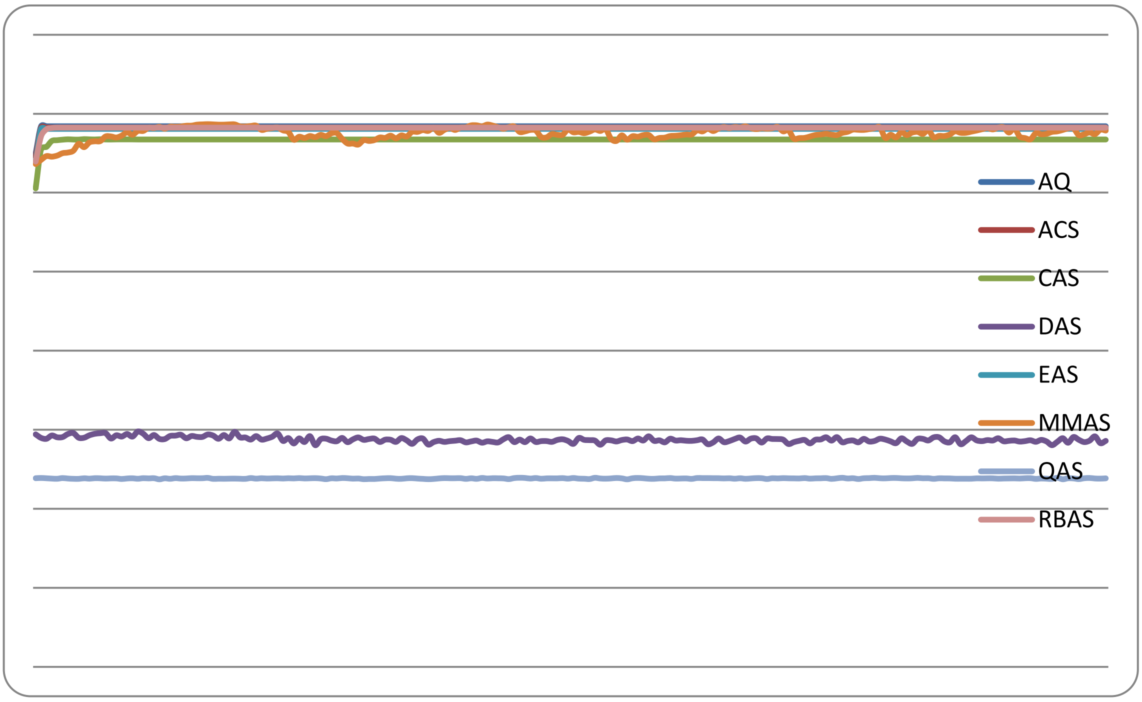


Rys. 1. Średnie czasy wykonywania poszczególnych algorytmów
Źródło: Krajewski, 2007.

Najlepsze rozwiązania odnajduje algorytm MMAS. Jeśli mamy zatem wystarczającą ilość czasu, jest to algorytm najlepszy. Jeśli jednak chcemy przyspieszyć uzyskanie rozwiązania – najlepszy okaże się algorytm AS_{rank}, który generuje prawie tak dobre wyniki, ale w o wiele krótszym czasie.

3.3. Uzasadnienie uzyskanych wyników

Obserwując wykres dla średniej znalezionej wartości widać algorytmy znajdujące dosyć szybko rozwiązanie poprawne; kierują wszystkie mrówki na prawidłowe „tory”. Algorytmy, które gorzej radzą sobie z rozwiązywaniem tego problemu próbują szukać innych, alternatywnych ścieżek, jednak najprawdopodobniej poszukiwania są zawężone do tak małego przedziału że nie udaje im się „wyrwać” z obszaru pierwszych znalezionych rozwiązań.



Rys. 2. Średnia jakość rozwiązań znalezionych przez różne algorytmy zastosowane w problemie plecakowym
Źródło: Krajewski, 2007.

4. PODSUMOWANIE

Algorytmy mrówkowe okazały się skutecznym narzędziem do rozwiązywania tak złożonego problemu, jak problem plecakowy. Porównanie algorytmów wskazuje te z nich, które uzyskują najlepsze rozwiązania i te, które robią to najszybciej. W pracy wskazano też te z algorytmów mrówkowych, które wykazują pożyteczne cechy, jakimi są intensyfikacja i dywersyfikacja. Żaden jednak z badanych algorytmów nie dominuje nad pozostałymi. Trzeba więc uzależniać typ zastosowanego algorytmu od wielkości problemu, czasu możliwego do wykorzystania, oczekiwań co do jakości uzyskanego rozwiązania.

Oddzielnym problemem jest wrażliwość algorytmów na wartości parametrów w nich zastosowanych. Wstępne badania wskazują, że można tu upatrywać drogi do poprawy ich działania, zarówno jakości znajdowanych rozwiązań, jak i czasu uzyskiwania satysfakcjonujących rezultatów.

LITERATURA

Boryczka, U. (2006). *Algorytmy optymalizacji mrowiskowe*. Katowice: Wydawnictwo Uniwersytetu Śląskiego.

Dorigo, M., Stutzle, T. (2004). *Ant Colony Optimization*. London: Massachusetts Institute of Technology.

Jankowiak, M. (2003). *Zastosowanie algorytmów mrówkowych do rozwiązywania symetrycznego problemu komiwojażera*. Software 2.0, 2/2003.

Krajewski, T. (2007). *Porównanie skuteczności wybranych algorytmów mrówkowych*. Łódź: Politechnika Łódzka.

Linda, T., Abrich, D. (2001). *Optymalizacja kolonii mrówek w zastosowaniu do problemu komiwojażera*. Wrocław: Wydział Informatyki i Zarządzania Politechniki Wrocławskiej.

Mills, P. (2000). *Monitors for GLS and other Meta-Heuristics*. Essex: Department of Computer Science, University of Essex.