

mgr inż. Arkadiusz Antczak
mgr inż. Paweł Antczak
dr hab. inż. Tadeusz Witkowski
Wydział Inżynierii Produkcji Politechniki Warszawskiej

ALGORYTM SYMULOWANEGO WYŻARZANIA - EFEKTYWNA METODA OPTYMALIZACJI HARMONOGRAMÓW PRODUKCJI

W pracy przedstawiono podstawowe charakterystyki procedury symulowanego wyżarzania do opracowania harmonogramów produkcji wieloasortymentowej. Omówiono procedurę symulowanego wyżarzania i przedstawiono przykłady jej zastosowania.

THE SIMULATED ANNEALING ALGORITHM - EFFICIENT METHOD OPTIMIZATION OF THE JOB SHOP SCHEDULING PROBLEM

The paper presents the characteristics of the simulated annealing procedure for optimization of the job shop scheduling problems (multi-assortment production). This paper overviews simulated annealing procedure and discusses an illustrative case study.

1. WPROWADZENIE

W badaniach operacyjnych wykorzystano wiele dokładnych modeli do harmonogramowania produkcji małoseryjnej. Jednak większość z nich zawiodła w praktyce ponieważ mamy tu do czynienia z problemem „NP- trudnym”. Proste metody heurystyczne takie, jak reguły priorytetu (reguła najkrótszej operacji, reguła pierwsze przyszło-pierwsze obsłużone) są szybkie, lecz jakość ich rozwiązania jest umiarkowana. Ostatnio opracowano bardziej efektywne metody takie jak sztuczne sieci neuronowe, algorytmy genetyczne, metoda Tabu oraz symulowanego wyżarzania (SA). Są one zdolne do generowania przybliżonych do optymalnych rozwiązań w stosunkowo umiarkowanym czasie obliczeń. W naszych wcześniejszych pracach [3,13,14] wykorzystaliśmy metody optymalizacji harmonogramowania bazujące na sztucznych sieciach neuronowych, algorytmach genetycznych, czy też metodach przesuwego wąskiego gardła.

W pracy w części drugiej podano krótkie wprowadzenie do optymalizacji lokalnej i globalnej, a w części trzeciej przedstawiono jedną z efektywniejszych metod optymalizacji kombinatorycznej - algorytm symulowanego wyżarzania. Część czwarta omawia podstawowe parametry tego algorytmu, natomiast w części piątej opisano zastosowanie algorytmu symulowanego wyżarzania w połączeniu z innymi metaheurystykami do procesu hybrydyzacji oraz zrównoleglenia obliczeń. Przedstawiono hybrydę algorytmu SA i algorytmów genetycznych AG oraz algorytm symulowanego wyżarzania dla systemów wieloagentowych.

2. OPTYMALIZACJA LOKALNA I GLOBALNA

O zadaniach optymalizacji globalnej mówi się wówczas, gdy funkcja celu ma w obszarze dopuszczalnym więcej niż jedno minimum lokalne. Wielość minimów lokalnych stanowi utrudnienie w konstruowaniu algorytmu optymalizacji – nie może on polegać na lokalnych właściwościach funkcji celu, a kolejne modyfikacje poprawiające nie zawsze prowadzą do rozwiązania globalnie optymalnego. Znanych jest wiele metod optymalizacji globalnej; dają one najczęściej rozwiązania przybliżone.

Wiele metod polega na modyfikacji algorytmu optymalizacji lokalnej. Należą do nich metody wielostartowe z modyfikacją funkcji celu, polegającą na „zasypywaniu” obszarów przyciągania minimów lokalnych znalezionych w kolejnych uruchomieniach algorytmu losowego. Inne metody bazują z kolei na symulacji ruchu punktu materialnego, który ma pewną bezwładność. Dodanie bezwładności sprawia, że oprócz ruchu w kierunku minimum lokalnego, punkt ma także składową „szybkości” wynikającą z ruchu wykonywanego w poprzednim kroku, dzięki czemu jest możliwe opuszczenie „płtykich” minimów funkcji celu. Metody tego typu charakteryzują się tym, że naraz przetwarzany jest tylko jeden punkt roboczy. Są również metody populacyjne, które badają całą populację punktów. Należą do nich metody bazujące na grupowaniu, w których wykonuje się naprzemiennie optymalizację lokalną i grupowanie punktów. Zakłada się, że punkty znajdujące się w otoczeniu tych samych minimów lokalnych będą w kolejnych krokach optymalizacji lokalnej zbliżać się do siebie. Identyfikacja takich grup pozwala usunąć część punktów z dalszych rozważań, gdyż ich istnienie przyczynia się co najwyżej do zwiększenia obliczeń.

Do rozwiązania zadań dyskretnych stosuje się metody dokładne, przybliżone i losowe [5]. Metody dokładne gwarantują wyznaczenie rozwiązania. Najprostszą, a zarazem najbardziej nieefektywną metodą, jest algorytm pełnego przeglądu. Jednakże do zadań o dużych rozmiarach jest on nie do przyjęcia ze względu na dużą złożoność obliczeniową rozwiązywanego problemu. Metody przybliżone wyznaczają przybliżone rozwiązania i pozwalają na określenie dokładności przybliżenia. Algorytmy aproksymacyjne, znane dla niektórych problemów, pozwalają ograniczyć błąd wyznaczenia wartości funkcji celu w minimum globalnym. Natomiast metody k -opt gwarantują, że w otoczeniu rozwiązania o promieniu k nie można znaleźć innego minimum lokalnego. Metody losowe nie dają gwarancji uzyskania rozwiązania, umożliwiają natomiast odnalezienie rozwiązania z pewnym prawdopodobieństwem, najczęściej rosnącym wraz z liczbą kroków metody. Istotą tych metod jest losowe próbkowanie przestrzeni rozwiązań.

Metody niedeterministyczne charakteryzują się wykorzystaniem losowości zarówno w procesie poszukiwań nowych rozwiązań, jak też w procesie przechodzenia między kolejnymi rozwiązaniami roboczymi. Najprostszą jest metoda Monte Carlo, gdzie z rozkładem jednostajnym dokonuje się szeregu losowań punktu z przestrzeni rozwiązań, pamiętając najlepszy dotychczas znaleziony. W metodzie błędzenia przypadkowego przetwarzają się jeden punkt roboczy. Kolejnym punktem roboczym, zostaje punkt niezależnie od wartości funkcji celu, przy czym pamiętane jest rozwiązanie „najlepsze”. Algorytm SA jest modyfikacją błędzenia przypadkowego, w której nowo wygenerowany punkt nie zawsze staje się rozwiązaniem roboczym. Jest nim wówczas, gdy poprawia on wartość funkcji celu, natomiast w przeciwnym przypadku, akceptacja następuje z pewnym prawdopodobieństwem.

3. ALGORYTM SYMULOWANEGO WYŻARZANIA

Algorytm SA (ang. simulated annealing) wziął swój początek od pracy W. Metropolis'a i inn. [1] (stąd często używana nazwa Schemat Metropolis'a), w której użyto jej do opisu ruchów termicznych atomów cieczy w ustalonej temperaturze. Algorytm ten działa na założonej konfiguracji cząstek. Krok obliczeniowy polega na losowym przesunięciu jednej cząstki i wyliczeniu zmiany energii całkowitej systemu spowodowanej tym przesunięciem. Przesunięcie zmniejszające wartość energii całkowitej jest bezwzględnie akceptowane, natomiast przesunięcie zwiększające tę energię może zostać zaakceptowane z określonym prawdopodobieństwem. Tak więc SA jest stochastyczną heurystyczną metodą optymalizacji, związaną z fizycznym procesem krystalizacji. Schemat Metropolis'a, po adaptacji dokonanej przez Kirkpatrick'a i inn. [8], znalazł szerokie zastosowania do rozwiązywania problemów optymalizacji kombinatorycznej. W tym przypadku rozwiązanie w przestrzeni rozwiązań dopuszczalnych jest odpowiednikiem stanu układu fizycznego S , funkcja celu odpowiada energii systemu E , a temperatura T jest parametrem kontrolno-sterującym. Technika ta jest podobna do tradycyjnych metod „gradientu spadku”, lecz zawiera składnik stochastyczny, aby zapobiec utknięciu w minimum lokalnym. Składnik ten jest kontrolowany przez parametr zewnętrzny zwany temperaturą T . Kiedy temperatura jest wysoka, wpływ stochastyczny jest silny, lecz wraz ze spadkiem temperatury składnik stochastyczny staje się coraz mniej ważny; tak więc proces stopniowo zmienia się z zachowania stochastycznego w normalne zachowanie się „gradientu spadku”. Znaczącą cechą wyżarzania symulowanego jest istnienie oficjalnego dowodu zbieżności.

Algorytm SA może być rozpatrywany w kilku etapach [7]:

1. Utworzyć początkowe rozwiązanie;
2. Ocenić to rozwiązanie;
3. Zmienić rozwiązanie w sposób losowy;
4. Ocenić nowe rozwiązanie;
5. Zmniejszyć temperaturę T . Jeżeli wartość T nie jest równa zero powrócić do pkt. 3.

Dla większości problemów początkowe rozwiązanie wybierane jest losowo. Na początku ono ma postać bieżącego rozwiązania (*current solution*). Inna możliwość polega na tym, że jako rozwiązanie początkowe przyjmuje się rozwiązanie otrzymane w wyniku wstępnego przeszukiwania.

Poszukiwanie rozwiązania rozpoczyna się z skopiowania bieżącego rozwiązania do postaci rozwiązania roboczego (*working solution*). Ta więc przeprowadza się modyfikację rozwiązania roboczego. Sposób modyfikacji rozwiązania roboczego zależy od tego w jaki sposób jest ono reprezentowane (kodowane). Np. w przypadku zadania szeregowania, każdy element może przedstawiać operacje. Aby wykonać poszukiwanie na podstawie rozwiązania roboczego, wybiera się np. dwa elementy i przedstawia je. Pozwala to na zachowanie całości rozwiązania, ponieważ przy tym nie występuje powtórzenie lub opuszczenie jakiegoś elementu (operacji). Po wykonaniu poszukiwania rozwiązania roboczego przeprowadza się ocenę rozwiązania. Poszukiwanie nowego rozwiązania przeprowadza się na ogół za pomocą metody Monte Carlo.

Na etapie określenia funkcji celu posiadamy dwa rozwiązania. Pierwsze – to oryginalne rozwiązanie (rozwiązanie bieżące), a drugie – znalezione nowe rozwiązanie (rozwiązanie robocze). Z każdym rozwiązaniem związana jest określona energia, przedstawiająca jego efektywność. Jeśli nowe rozwiązanie jest lepsze od pierwotnego wówczas będzie ono przyjęte. W tym przypadku kopiujemy rozwiązanie robocze do postaci rozwiązania bieżącego i przechodzimy do obniżenia temperatury. Jeśli jest ono

gorsze niż rozwiązanie bieżące - może być również przyjęte, lecz tylko z prawdopodobieństwem p równym:

$$p = \frac{1}{1 + e^{\frac{|E_{old} - E_{new}|}{T}}}$$

gdzie E_{old} i E_{new} - odpowiednie wartości funkcji celu; p zależy od temperatury T .

Przy wysokiej T słabe rozwiązania są akceptowane częściej, niż odrzucane. Jeżeli energia jest mniejsza, prawdopodobieństwo przyjęcia rozwiązania jest wyższe. Przy obniżeniu T prawdopodobieństwo przyjęcia gorszego rozwiązania także obniża się. Przy tym wyższy poziom energii także wpływa na zmniejszenie prawdopodobieństwa przyjęcia gorszego rozwiązania. Przy wysokich temperaturach SA pozwala przyjmować gorsze rozwiązania w tym celu aby przeprowadzić pełniejsze przeszukiwanie rozwiązań. Algorytm po wielu iteracjach z daną T nieznacznie ją zniża. Efektywna procedura symulowanego wyżarzania wymaga określenia m.in. sposobu generacji sąsiedztwa, efektywnego algorytmu schładzania, czy też szybkiej metody oszacowania rozwiązań sąsiednich.

4. STEROWANIE PROCESEM SYMULOWANEGO WYŻARZANIA

4.1. Sąsiedztwo rozwiązań

Jakość rozwiązania uzyskanego za pomocą metod przeszukiwania lokalnego zależy w dużym stopniu od sposobu zdefiniowania sąsiedztwa N . Zdefiniowanie N powoduje jednocześnie ukształtowanie przestrzeni przeszukiwań i określenie w nim strategii sterowania wyborem rozwiązań sąsiednich. Do głównych cech, które powinny charakteryzować sąsiedztwo zalicza się niżej wymienione [10].

Korelacja: rozwiązanie sąsiednie powinno być w dużym stopniu skorelowane z rozwiązaniem pierwotnym, czyli sposób tworzenia N powinien zapewniać, że należące do niego rozwiązania różnią się jedynie nieznacznie od rozwiązania pierwotnego. Cecha ta umożliwia dokładniejsze przeszukiwanie przestrzeni rozwiązań.

Tworzenie rozwiązań dopuszczalnych: zmiany w rozwiązaniu pierwotnym powinny zawsze prowadzić do powstawania rozwiązań dopuszczalnych. W przypadku przeciwnym, poszukiwania powinny się ograniczyć do dziedziny rozwiązań dopuszczalnych.

Poprawa: sposób tworzenia N powinien zagwarantować możliwość powstawania w jego ramach rozwiązań lepszych od rozwiązania pierwotnego.

Wielkość sąsiedztwa: przeciętna wielkość N powinna zawierać się w pewnych dopuszczalnych granicach. Mała liczba możliwych ruchów w ramach sąsiedztwa może doprowadzić do zatrzymania procesu optymalizacji przez stosunkowo słabe optima lokalne już w początkowej fazie. Natomiast, duża liczba możliwych ruchów może wpłynąć na znaczne spowolnienie procesu optymalizacji, z powodu konieczności wykonania dużej liczby obliczeń.

Łączność: powinna być zagwarantowana droga składająca się ze skończonej liczby kroków, łącząca rozwiązanie dane z optymalnym. W skład drogi mogą wchodzić kroki polepszające i pogarszające rozwiązanie; przy czym te pogarszające rozwiązania są

często nieuniknione. Jeżeli sąsiedztwo nie ma cechy łączności, to może się zdarzyć, że pewne obszary zostaną całkowicie wyłączone z poszukiwań.

Dla problemu kombinatorycznego typu $m \times n$ (w przypadku harmonogramowania m maszyn oraz n części) istnieje kilka definicji sąsiedztwa [10].

Pierwsze najprostsze sąsiedztwo $N_1(H)$ określono następująco. Zdefiniowanie sąsiedztwa polega na dowolnej zmianie kolejności wykonywania operacji na maszynie M_k . Wadą tego sąsiedztwa jest m.in. to, że dowolna zmiana kolejności wykonania operacji na maszynie może spowodować powstanie rozwiązania niedopuszczalnego. Kolejną wadą jest to, że rozmiary tego sąsiedztwa są zbyt duże. Tak np. w przypadku problemu $m \times n$, jeżeli każda praca jest wykonywana na każdej maszynie, jego wielkość wynosi $m(n-1)$. Niektóre z tych wad mogą być usunięte, przez ograniczenie zmian tylko do operacji, które występują na maszynach bezpośrednio po sobie. Na ogół dla danego harmonogramu sąsiedztwo $N_1(H)$ składa się ze wszystkich harmonogramów otrzymanych z H przez zmianę łuku ($v \rightarrow w$) należącego do ścieżki H_k . Przynajmniej jeden z elementów v i w jest pierwszym lub ostatnim elementem bloku. Przy czym blokiem jest łańcuch kolejnych operacji na ścieżce krytycznej, które są wykonywane na tej samej maszynie, a ścieżka krytyczna określa wielkość C max tj. termin, w którym kończy się realizacja wszystkich prac. Do pierwszego bloku rozpatrywane są jedynie v i w leżące na jego końcu, a dla ostatniego bloku rozpatrywane są jedynie v i w leżące na jego początku.

Drugie sąsiedztwo $N_2(H)$ określa możliwość zamiany miejscami operacji, które nie są realizowane bezpośrednio po sobie i są położone wewnątrz bloku. Umożliwia to „zapełnianie luk” we wcześniej zbudowanym harmonogramie. Sąsiedztwo $N_2(H)$ można zdefiniować następująco. Niech operacja v będzie elementem bloku, takim że $b = (b', v, b'')$. Przy tworzeniu sąsiedniego rozwiązania v jest przenoszony bliżej do pierwszej lub ostatniej operacji w ramach bloku tak, aby w wyniku takiego działania powstało rozwiązanie dopuszczalne. Niedopuszczalność rozwiązania objawia się powstaniem konturu w sieci odwzorowującej rozwiązanie. Jeżeli przeniesienie operacji z wnętrza bloku na miejsce początkowe lub ostatnie nie jest możliwe, to należy przenieść operację w inne miejsce jak najmniej oddalone od miejsca pierwszego i ostatniego.

Sąsiedztwo $N_3(H)$ jest rozszerzeniem sąsiedztwa pierwszego przez uwzględnienie możliwości dokonania zmiany kierunku dwóch łuków w jednym kroku. Zmiany polegające na zmianie kierunku dwóch łuków mogą spowodować poprawę rozwiązania, zaś w szczególnych przypadkach zmiana kierunku jednego łuku nie prowadzi do poprawy rozwiązania. Sąsiedztwo $N_3(H)$ można zdefiniować następująco. Niech v i w będą kolejnymi operacjami należącymi do ścieżki krytycznej. Wszystkie możliwe permutacje elementów $\{PM_v, v, w\}$ lub $\{v, w, SM_w\}$ są określane jako sąsiednie, jeżeli zmieniana jest także kolejność elementów v i w . PM_v jest to operacja poprzedzająca operację v na maszynie M_k , zaś SM_w jest to następnik operacji w na maszynie M_k .

Czwarte sąsiedztwo $N_4(H)$ zawiera w sobie wszystkie inne wcześniej opisane sąsiedztwa. Z wielu badań wynika, że za jego pomocą można uzyskać bardzo dobre wyniki.

4. 2. Schemat schładzania

Ważnym problemem w przypadku użycia symulowanego wyżarzania jest dobór sposobu zmiany T_k . Procedurę tą określa się mianem schematu (harmonogramu) schładzania. Temperatura T_k jest obniżana w miarę upływu czasu w celu wykluczenia

lub prawie uniemożliwienia „błędnych ruchów” w końcowym okresie poszukiwań. W klasycznym harmonogramie obniżania wartości T_k proces ten rozpoczyna się od wartości T_0 . Następnie temperatura jest utrzymywana na stałym poziomie w eksperymencie przez pewną liczbę kroków algorytmu. Potem jest ona obniżana przez pomnożenie jej przez stały współczynnik ($0 < \alpha < 1$) po każdej serii L kroków. Stąd po kL krokach temperatura wynosi $T(kL) = T_k = \alpha^k T_0$.

Istnieje wiele wariantów obniżania temperatury. I tak w eksperymencie [2] schemat schładzania różni się w pewnym stopniu od podejścia klasycznego. Harmonogram schładzania w podejściu klasycznym zmniejsza temperaturę stopniowo, aż do schłodzenia procesu, kiedy algorytm się kończy. W tym przypadku stosuje się podejście w którym, w momencie kiedy proces staje się schłodzony temperatura zostaje obniżona do poziomu początkowego, a bieżące rozwiązanie jest porównywane z najlepszym rozwiązaniem znalezionym do danego momentu. Proces schładzania i ponownego nagrzewania powtarzany jest wielokrotnie. Podejście to posiada efekt pozwalający na zbadanie w procesie SA szerokiego spektrum rozwiązań wokół rozwiązania bieżącego, a jeśli w procesie występuje tendencja do odchyień, sprowadza go do najlepszego znalezionego rozwiązania.

5. ZRÓWNOLEGIENIE I HYBRYDYZACJA ALGORYTMU SA

W ostatnich latach [2, 5, 6, 9, 12] badacze coraz częściej wykorzystują w celu optymalizacji obliczeń komputerowych podejścia do zrównoleglenia i hybrydyzacji poszczególnych metaheurystyk m.in. symulowanego wyżarzania, algorytmów ewolucyjnych itp.

Algorytmy ewolucyjne (AE) są technikami, które w szczególny sposób nadają się do zrównoleglenia. Wynika to z faktu, że naraz przetwarzanych jest wiele wzajemnie niezależnych osobników. O równoległości w algorytmach AE mówi się na etapie implementacji oraz koncepcji. W pierwszym przypadku, schemat algorytmu pozostaje bez zmian tj. w dalszym ciągu selekcja jest scentralizowana. Natomiast drugi wariant oznacza decentralizację selekcji – algorytmy takie są nazywane koewolucyjnymi. Zrównoleglenie AE na etapie implementacji jest możliwe dzięki temu, że na algorytm ten składa się z wielu, niezależnych od siebie – podobnych elementów. Zasada koewolucji polega na tym, że globalne oddziaływanie na poziomie selekcji i krzyżowania, znamienne dla standardowego AE, stają się w pewnym stopniu lokalne. Zasadę koewolucji realizuje się dwoma sposobami: za pomocą algorytmu wyspowego (podpopulacyjnego) oraz algorytmu komórkowego (masowo równoległego) [2].

Algorytm wyspowy składa się z wielu populacji, które ewoluują niezależnie od siebie, sporadycznie wymieniając między sobą informacje. Koewolujące populacje mogą się składać z osobników tego samego typu bądź też różnych typów. W każdej z podpopulacji wykonywany jest algorytm ewolucyjny. Wykonywany w podpopulacji AE tym różni się od schematu z jedną populacją, że dopuszcza dopływ osobników z zewnątrz. Możliwe jest też wysyłanie osobnika na zewnątrz, przy czym osobniki te giną, lecz są wprowadzane do innych podpopulacji. Wymiana osobników między podpopulacjami odbywa się za pomocą dwóch podstawowych modeli: emigracji i imigracji. Emigracja polega na tym, że osobnik wysyłany na zewnątrz jest usuwany z populacji bazowej. Podczas imigracji natomiast, na zewnątrz wysyłana jest tylko kopia

osobnika, on sam pozostaje w populacji bazowej. Wymiany informacji między podpopulacjami można dokonywać różnymi sposobami – zbiór podpopulacji może być nieuporządkowany albo też może być przyjęta relacja sąsiedztwa podpopulacji. W pierwszym przypadku wymiana osobników następuje między dowolnymi dwiema podpopulacjami, w drugim zaś jest ograniczona do sąsiadujących podpopulacji.

Algorytm komórkowy charakteryzuje się tym, że w populacji osobników obowiązuje pewna arbitralnie wybrana topologia (nie związana z topologią przestrzeni genotypów). Dla każdego osobnika X_i jest zdefiniowane sąsiedztwo $N_r(X_i)$ (r jest promieniem sąsiedztwa) i każdy osobnik podlega niezależnej ewolucji. Ewolucja polega na tym, że osobnik wchodzi w interakcje (selekcja, krzyżowanie i mutacja) z osobnikami z $N_r(X_i)$, co prowadzi do wygenerowania nowego osobnika Y , który konkuruje z X_i na etapie sukcesji (tworzy się nową populację bazową, mogącą zawierać osobniki zarówno z populacji potomnej, jak i ze starej populacji bazowej). Algorytm komórkowy nazywany również dyfuzyjnym z powodu sposobu rozprzestrzeniania się rozwiązania, które powstało w miejscu jednego z osobników populacji. W kolejnych generacjach rozwiązanie to, wskutek działania selekcji, powiela się dla osobników sąsiednich.

Hybrydyzacja algorytmów AE z algorytmami SA stanowi jeden z bardziej efektywnych sposobów optymalizacji kombinatorycznej w tym harmonogramowania. W [11] przedstawiono podejście hybrydowe obejmujące algorytm SA oraz AG (algorytm genetyczny). Podczas gdy algorytm SA przechodzi od jednego punktu przestrzeni rozwiązań do innego, algorytmy genetyczne wiążą się z wieloma punktami równocześnie. Punktem kluczowym zastosowania algorytmów genetycznych w harmonogramowaniu, jest wybór właściwej reprezentacji i wykorzystanie określonych operatorów mutacji i skrzyżowania. W [12] zmodyfikowano podejście opisane przez Nakano i Yamadę dla małego rozmiaru zadania. Ich przedstawienie bazuje na relacjach między parami zadań. AG zaczyna się od generowania określonej liczby osobników będących populacją początkową. W przypadku gdy jest ona przeprowadzana losowo, niektóre osobniki mogą reprezentować rozwiązania niedopuszczalne. Za zgodność i naprawę rozwiązań odpowiadają dwa algorytmy nazywane lokalnym i globalnym harmonizatorem. Następnie jest obliczane „dopasowanie” wszystkich osobników i przeprowadza się operacje „genetyczne”. Osobniki są selekcjonowane do mutacji i krzyżowania. Zanim nowa populacja będzie utworzona, zmodyfikowane osobniki muszą utworzyć harmonogram. Następnie ich „dopasowanie” jest obliczane itd. Proces ten jest kontynuowany dopóki populacja dąży do wystarczająco dobrej wartości „dopasowania”.

W [9] rozpatrzono równoległe rekombinacyjne wyżarzanie symulowane PRSA. Zarówno SA jak i AG są silnymi technikami optymalizacyjnymi. Niektóre cechy mają wspólne, a niektóre różniące się. Obie metody są iteracyjne zaczynając od pewnych wyjściowych rozwiązań. Zaburzają one rozwiązania w pewien sposób aby znaleźć lepsze rozwiązania (mówiąc innymi słowami obie techniki generują nowe punkty w przestrzeni poszukiwań poprzez zastosowanie operatorów do bieżących punktów). Cechy różniące obie metody to fakt, że SA rozpatruje jedno rozwiązanie na raz, podczas gdy AG zawiera pojęcie populacji i rozpatrują kilka rozwiązań jednocześnie. Dlatego też AG w odróżnieniu do SA wykazują wyraźną równoległość. Ryzyko tego, że SA „utknie” w lokalnym minimum można zredukować poprzez przeprowadzenie algorytmu kilka razy, lub przez przeprowadzenie więcej niż jednego algorytmu na różnych

procesorach równoległego komputera. W odróżnieniu od AG, SA posiada zewnętrzny mechanizm sterujący - temperaturę. Wpływ temperatury jest początkowo bardzo silny lecz stopniowo się zmniejsza. Jedną z zalet SA jest fakt, że istnieje formalny dowód zbieżności, podczas gdy zbieżność AG nie może być zapewniona. Inna różnica polega na tym, że AG posiadają dobrą „pamięć”. W momencie gdy wykorzystują dużą ilość rozwiązań na raz, mogą one użyć całej swej „wiedzy” do dalszych poszukiwań. W przypadku SA jest tylko jedno rozwiązanie dostępne na raz, więc poprzednie informacje mogą być utracone i nie odzyskane.

W PRSA łączą się korzystne cechy z obu metod: właściwość zbieżności SA i właściwość równoległości AG. Podczas gdy w „normalnych” AG tylko nowe osobniki o lepszym dopasowaniu są akceptowane, tutaj te gorsze także mogą być zaakceptowane z pewnym prawdopodobieństwem. Proces generowania podpopulacji jest powtarzany aż wszystkie jednostki zostaną wyselekcjonowane do krzyżowania dokładnie jeden raz. Później temperatura jest obniżana o małą wartość i zaczyna się tworzenie nowej populacji.

Kryterium zakończenia może być statyczne, dynamiczne lub statyczne i dynamiczne.. Kryterium statyczne bazuje np. na temperaturze, a kryterium dynamiczne bazuje na stanie procesu rozwiązania i może być zaadoptowane z algorytmów genetycznych, np. liczba iteracji bez znaczących zmian populacji.

Jakość wyników uzyskanych przy PRSA a także przy jego wdrożeniu równoległym, zależy od kilku parametrów, które są nastawiane zewnętrznie, takich jak schładzanie, rozmiar populacji, strategia selekcji, współczynnik migracji i strategia migracji.

Schemat schładzania ma znaczący wpływ na zbieżność i jakość rozwiązania. Jeśli temperatura spada szybko wówczas algorytm zbiega się szybko, lecz wyniki mają tendencję do pogarszania się. Z drugiej strony powolne schładzanie spowalnia zbieżność algorytmu, lecz daje lepsze wyniki.

Jeśli rozmiar populacji jest bardzo duży, wówczas są szanse, że PRSA zbiegnie się wcześniej z dobrymi wynikami. Jednak wymaga to dużej ilości pamięci i czasu CPU na obliczenie jednej populacji. Jeśli rozmiar populacji jest mały wymagana jest mniejsza pamięć i czas obliczania populacji będzie krótki, lecz może „zająć wiele pokoleń” zanim zbadamy przestrzeń rozwiązań i otrzymamy dobre rozwiązanie.

Wskaźnik migracji (ilość emigrantów wystanych z jednego węzła) wpływa na zbieżność w następujący sposób: jeśli duża ilość jednostek jest wysłana/odebrana wówczas wszystkie lokalne populacje mają tendencje do upodabniania się. Tak więc PRSA może zbiegać się szybko lecz rozwiązanie będzie prawdopodobnie tylko umiarkowane. Szybkość przetwarzania także będzie niska gdyż komunikacja – wysyłanie wielu jednostek – zużyje mnóstwo czasu. Z drugiej strony, jeśli wskaźnik migracji jest niski wówczas wszystkie procesory będą szukały mniej lub bardziej niezależnie i prawdopodobnie nie znajdą globalnego minimum w sensownym czasie. Migracja większej ilości jednostek może skierować poszukiwania w nowe rejony i w ten sposób pomóc w skierowaniu przeszukiwania w stronę najlepszego rozwiązania.

W [12] zaimplementowano algorytm SA w rozproszonym środowisku obliczeniowym na komputerze Linux PC – cluster z 12 procesorami. Każdy procesor uruchamia swój własny proces SA w niezależnym wyjściowym rozwiązaniu. Strategia komunikacji jest używana kiedy indywidualne obliczenia są przerwane i temperatura jest M razy obniżona. Eksperymenty dają stabilne wyniki równe lub bliskie do najlepszych

rozwiązań dla zadań małego rozmiaru. Jednak w tym przypadku wybór parametru M nie ma znaczącego wpływu na wartość funkcji celu oraz czas obliczeń. W porównaniu do obliczeń szeregowych (1 procesor) przy obliczeniach równoległych osiągnięto przyspieszenie obliczeń do około 10 razy.

W [6] przedstawiono algorytm ewolucyjny symulowanego wyżarzania (ESA), który może być zrównolegniony w postaci różnych wielostartowych niezależnych przebiegów (Multiple Independent Runs - MIR). W przypadku wykorzystania do zrównoleglenia obliczeń metody MIR, uruchamiany jest ten sam algorytm jednocześnie, przy czym każdy wykonuje inny proces z różnymi rozwiązaniami wyjściowymi. W tym przypadku agenci (wyspy) niezależnie uruchamiają ESA. Jeżeli uruchomimy ten algorytm na równoległej maszynie takiej jak MIMD (wielokrotny strumień rozkazów – wielokrotny strumień danych), wtedy można przeprowadzić zrównoleglenie w różny sposób. Jednym z zadań systemu wieloagentowego na komputerach równoległych jest problem skalowania, który związany jest z zagadnieniem dekompozycji zadania i przydzielenia go do odpowiedniego procesora. Do uruchomienia rozproszonego algorytmu ewolucyjnego wyżarzania symulowanego (dASA) wymagane jest zrównolegnięte środowisko obliczeniowe. Infrastrukturą która zabezpiecza środowisko do rozwiązywania rozproszonego problemu wykorzystującego podejście wieloagentowe jest maszyna o rozproszonych zasobach (Distributed Resource Machine-DRM). Jest ona rozproszoną infrastrukturą oprogramowania DREAM, które było opracowane do rozwiązywania problemów przez rozproszone algorytmy ewolucyjne. Głównym celem tego systemu jest rozwiązywanie problemów opartych na systemach wieloagentowych, które uruchamiają algorytmy ewolucyjne.

W [6] przedstawiono zależność czasu pracy CPU od liczby agentów, podczas badania trzech zadań o różnym rozmiarze ($n \times m$): A – stosunkowo małym, B – średnim, C – dużym. W eksperymentach wykorzystano 4 różne wielkości agentów: 5, 10, 15 i 20. Badano zmiany czasu CPU dla zadań małych A , średnich B oraz dużych C . Ponieważ zadanie A posiada mały rozmiar, w porównaniu z dwoma pozostałymi B i C nie zauważa się znacznej zmiany w zużyciu czasu CPU. Z drugiej strony zmiana jest istotna w przypadku zadania C , gdyż w tym przypadku CPU pracuje około 15 000 s. (5 agentów) i zmniejsza się do 6 000 s. (20 agentów), a więc 2,5 krotnie. W przypadku zadania B , czas pracy CPU zmniejsza się z 5920 s. do 3560 s. tj. ponad 1,5 krotnie, przy czym przy wzroście liczby agentów powyżej 5-ciu nie zaobserwowano znaczącego zmniejszenia czasu pracy CPU. Można zauważyć że algorytm SA potrzebuje pewnego minimalnego czasu w celu osiągnięcia wartości optymalnej i ten czas jest wymagany nawet jeżeli zastosuje się masowe zrównoleglenie obliczeń. Można założyć, że jeżeli nastąpi zwiększenie liczby agentów dla realizacji zadania C to również będzie można zauważyć tę tendencję. Wynika stąd, że zrównoleglenie obliczeń pomaga przyspieszyć proces SA, ale będzie on zatrzymany jeżeli liczba procesorów osiągnie pewien poziom.

6. WNIOSKI

Przeprowadzone eksperymenty dla pewnej klasy zadań harmonogramowania [4] o określonym rozmiarze problemu (około 200 operacji) pokazały dużą efektywność procedury SA. Analiza literatury pokazuje również dużą efektywność SA w połączeniu z innymi metodami, a także przy obliczeniach równoległych. Tak np. procedura PRSA jest znacznie szybsza i daje lepsze rozwiązania niż SA. Porównując z AG, PRSA jest

także lepsza lecz trwa dłużej. Równoległa wersja PRSA pokazała ponadto zarówno poprawę czasu obliczeń jak również wartości funkcji celu. Jednakże średnia poprawa jakości rozwiązania w porównaniu do SA była nieznaczna. Zrównoleglenie obliczeń z wykorzystaniem systemu wieloagentowego pomaga przyspieszyć proces SA, ale zatrzymuje się on jeżeli liczba agentów osiągnie określony poziom. Celowe są więc dalsze badania w tym zakresie do określenia efektywności poszczególnych algorytmów w zależności od rodzaju problemu, kryterium optymalności, czy też liczby procesorów.

LITERATURA

- [1] Aarts, E.H.L., Korst, J.H.M. Simulated Annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing; N. Y. 1989
- [2] Alba E. (red.), Parallel Metaheuristics. A New Class of Algorithms, John Wiley & Sons, New Jersey 2005.
- [3] Antczak A., Witkowski T., Antczak P. Analiza problemu harmonogramowania wieloasortymentowej produkcji małoseryjnej za pomocą procedury hybrydowej (heurystyki SB i GRASP), Automation 2005, PIAP, Warszawa 2005, s. 126-135.
- [4] Antczak P., Antczak A., Witkowski T. Zastosowanie algorytmu symulowanego wyżarzania do optymalizacji harmonogramowania produkcji małoseryjnej, Automation 2006, PIAP, Warszawa 2006.
- [5] Arabas J. Wykłady z algorytmów ewolucyjnych, WNT, Warszawa 2001.
- [6] Aydin M.E., Fogarty T.C. A Simulated Annealing Algorithm for Multi-Agents Systems: A Job Shop Scheduling Application, Journal of Intelligent Manufacturing, 2004, 15 (6), p. 805-814.
- [7] Jones M. Tim. AI Application Programming, Charles River Media, Hingham, 2003.
- [8] Kirkpatrick, S., Gelatt (Jr.), C. D., Vecchi, M. P. Optimization by Simulated Annealing, Science 220 (1983) 5, pp. 571-680.
- [9] Mahfoud, W. S., Goldberg, D. E. Parallel recombinative simulating annealing: a genetic algorithm, Technical Report No. 92002, University of Illinois, 1992.
- [10] Mattfeld D.C. Evolutionary Search and the Job Shop, Ph.-Verlag, Heidelberg 1996.
- [11] Nakano, R., Yamada, T. Conventional Genetic Algorithm for Job Shop Problems; in: Belew, R. K., Booker, L. B. (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA 1991, pp. 474-478.
- [12] Steinhofel K.A., Albrecht A., Wong C.K. Fast Parallel Heuristics for the Job Scheduling Problem, Computers on Operations Research, 29, p. 151-169.
- [13] Witkowski T., Antczak A., Antczak P. Random and Evolution Algorithms of Tasks Scheduling and the Production Scheduling, Proceedings of Internat. Joint Conf. on Fuzzy Systems-FUZZ-IEEE 2004, Budapest 2004, vol. 2, pp. 727-732.
- [14] Witkowski T., Antczak P., Strojny G. The Implementation of Neural Network for the Optimization of the Production Scheduling, Proceedings of III Internat. Conf. on Advances in Production Engineering – APE 2004, Warsaw 2004, part I, pp. 121-130.