



Politechnika Wrocławska

Wydział Informatyki i Zarządzania

kierunek studiów: Informatyka

specjalność: Inżynieria oprogramowania

Praca dyplomowa - magisterska

WIELOKRYTERIALNY PROBLEM
ROZMIESZCZENIA ZRASZACZY WODNYCH
NA ZADANEJ POWIERZCHNI
MULTICRITERIA WATER SPRINKLERS DEPLOYMENT PROBLEM
ON A GIVEN AREA

inż. Grzegorz Dziedzic

słowa kluczowe:
optymalizacja wielokryterialna,
algorytmy genetyczne,
zraszacze wodne

krótkie streszczenie:
SHORT ABSTRACT

Promotor:	dr Mariusz Fraś
	<i>imię i nazwisko</i>	<i>ocena</i>	<i>podpis</i>

Do celów archiwalnych pracę dyplomową zakwalifikowano do:*

a) kategorii A (akta wieczyste)

b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)

* niepotrzebne skreślić

pieczęć wydziałowa

Wrocław 2017

Niniejszy dokument został złożony w systemie L^AT_EX.

ABSTRACT PL

Streszczenie

ABSTRACT EN

Abstract

Spis treści

Rozdział 1. Wstęp	1
1.1. Wprowadzenie	1
1.2. Cel pracy	1
1.3. Przegląd literatury	2
1.4. Opis pracy	2
Rozdział 2. Problem nawodnienia obszaru	3
Rozdział 3. Optymalizacja	5
3.1. Optymalizacja jednokryterialna	5
3.2. Optymalizacja wielokryterialna	5
Rozdział 4. Algorytmy genetyczne	7
4.1. Opis ogólny	7
4.2. Algorytmy wielokryterialne	14
4.2.1. NSGA-II	15
4.2.2. SPEA	18
4.3. Strojanie	20
Rozdział 5. Systemy wspomagania decyzji	21
Rozdział 6. Rozwiązanie problemu	23
6.1. System wspomagania decyzji	23
6.1.1. Architektura	23
6.1.2. Interakcja z użytkownikiem	23
6.2. Optymalizacja	23
6.2.1. Model matematyczny	23
6.2.2. Porównanie algorytmów genetycznych	25
Rozdział 7. Podsumowanie	29
Dodatek A. Appendix 1	31

Rozdział 1

Wstęp

1.1. Wprowadzenie

Odpowiednie nawodnienie ogrodu jest jedną z podstawowych czynności pielęgnacyjnych. Gdy właściciel dysponuje odpowiednim budżetem najlepszym rozwiązaniem będzie dla niego inwestycja w automatyczny system nawadniania. System taki składa się z zraszaczy wodnych, rur pomiędzy nimi oraz systemu sterowania. Takie rozwiązanie pozwala zaoszczędzić czas tracony na ręcznym podlewaniu ogrodu oraz zapewnia równomierne nawodnienie na całej ustalonej powierzchni. Jednym z głównych problemów koniecznych do rozwiązania podczas instalacji takiego systemu jest odpowiednie rozmieszczenie poszczególnych zraszaczy. Te najczęściej znajdują się pod ziemią oraz posiadają wynurzalną głowicę. Z tego powodu raz zainstalowany zraszacz najczęściej zostaje na swoim miejscu, aż do momentu wymiany całej instalacji wodnej. Biorąc to pod uwagę rozmieszczenie zraszaczy powinno być dobrze przemyślane już podczas etapu projektowania systemu nawadniania. Projektując taki system należy przyjąć jako cel nawodnienie całości wskazanego obszaru jak najmniejszym kosztem przy przestrzeganiu wskazanych przez właściciela ograniczeń.

Proces projektowania sieci zraszaczy może być żmudny oraz długotrwały, biorąc pod uwagę różnorodność sprzętu dostępnego na rynku czy chociażby nieregularność powierzchni, która ma zostać nawodniona. Z pomocą może przyjść tutaj nowoczesna technologia. Opisany powyżej problem idealnie nadaje się do rozwiązania przy pomocy dostępnych algorytmów optymalizacyjnych. Praca ta będzie skupiać się na rozwiązaniu omówionego problemu poprzez opracowanie systemu wspomagania decyzji i implementację oraz porównanie wielokryterialnych algorytmów genetycznych.

1.2. Cel pracy

Opracowanie informatycznego systemu wspomagania podejmowania decyzji w rozmieszczeniu zraszaczy wodnych na zadanej powierzchni, a w szczególności zaproponowanie oraz przebadanie wielokryterialnych algorytmów genetycznych w kontekście wybranego problemu.

1.3. Przegląd literatury

Problem przedstawiony w temacie pracy nie był do tej pory poruszany w literaturze. Nie mniej jednak biorąc pod uwagę, opisane później, założenia przyjęte podczas realizacji pracy można znaleźć publikacje o tematyce zbliżonej, czyli takie w których autorzy starają się rozwiązać problem pokrycia danego obszaru (Area Coverage Problem).

W większości znalezionych publikacji do rozwiązania stawianego problemu używane są algorytmy ewolucyjne, w tym głównie algorytmy genetyczne. Przykładem jest "ŻEF", gdzie autorzy rozwiązują optymalizacyjny problem rozmieszczenia sensorów w sieci bezprzewodowej przy użyciu algorytmu genetycznego NSGA-II. Autorzy skupiają się na pokryciu określonego terenu sygnałem z jak najmniejszej ilości sensorów. Przedstawione wyniki są obiecujące - w 500 pokoleń ilość potrzebnych sensorów z BEGIN spadła do END. TODO

1.4. Opis pracy

W kolejnych rozdziałach opisane będą poszczególne zagadnienia związane z realizacją celu pracy. Najpierw dokładnie opisany zostanie problem nawodnienia obszaru, czyli problem z którym muszą zmagać się wszyscy projektanci ogrodów i systemów nawadniania. Następnie wyjaśnione zostaną pojęcia optymalizacji oraz optymalizacji wielokryterialnej, czyli dwa zagadnienia na których bazuje cała praca. W kolejnym rozdziale opisane zostaną algorytmy genetyczne, zarówno te w wersji podstawowej jak i wielokryterialnej wraz z ich wybranymi wersjami, tak aby przybliżyć czytelnikowi dlaczego i w jaki sposób one działają. W następnej kolejności zostanie wyjaśnione czym jest system wspomagania decyzji, jakie założenia powinien spełniać oraz jakimi funkcjonalnościami się wyróżniać. Kolejne rozdziały będą już odzwierciedleniem wykonanej pracy i dokładnym opisem przyjętego sposobu realizacji zadania. Najpierw dokładnie opisany zostanie opracowany system wspomagania decyzji. Zaprezentowana będzie architektura oraz przykładowe zrzuty ekranu pokazujące przygotowany interfejs graficzny. W kolejnym rozdziale skonkretyzowany zostanie problem optymalizacji. Przedstawiony będzie opracowany model matematyczny, przykładowe rezultaty oraz wyniki badań mające na celu porównania wybranych wersji algorytmów genetycznych. Na końcu pracy znajdzie się podsumowanie oceniające przedstawione rozwiązanie oraz propozycje dalszego rozwoju opracowanego systemu oraz badań.

Rozdział 2

Problem nawodnienia obszaru

Jednym z podstawowych problemów z jakim spotykają się właściciele ogrodów jest instalacja odpowiedniego systemu nawadniania. Mogą to zrobić sami albo zlecić zadanie firmie zajmującej się projektowaniem ogrodów i systemów nawadniania. Jest kilka szczególnie ważnych elementów, na które należy zwrócić uwagę projektując taki system:

- Odpowiedni pomiar terenu, który ma zostać nawodniony
- Wzięcie pod uwagę ukształtowania terenu
- Obliczenie potrzebnego ciśnienia wody
- Wybór i rozmieszczenie zraszaczy
- Wybór i poprowadzenie rur
- Umieszczenie zaworów

Wszystkie wymienione powyżej elementy znacząco wpływają na cenę, jakość oraz wydajność zaprojektowanego systemu.

W tej pracy autor skupia się na dwóch z powyższych punktów: wyborze i rozmieszczeniu zraszaczy oraz poprowadzeniu rur i nie będzie brał pod uwagę reszty wymienionych punktów.

Odpowiednie rozmieszczenie zraszaczy jest ważne, ponieważ zapewnia równomierne nawodnienia.

Rozdział 3

Optymalizacja

3.1. Optymalizacja jednokryterialna

3.2. Optymalizacja wielokryterialna

Rozdział 4

Algorytmy genetyczne

Problem optymalizacyjny opisany na wstępie pracy został rozwiązany przy użyciu algorytmów genetycznych. Są to algorytmy, których działanie oparte jest na tych samych zasadach na których odbywa się proces ewolucji zachodzący w naturze. W tym rozdziale przedstawiony zostanie opis ogólny algorytmu genetycznego, jego powiązanie z procesami zachodzącymi w przyrodzie, schemat działania oraz dokładny opis wykorzystywanych operatorów: krzyżowania, selekcji oraz mutacji. Następnie przedstawiony zostanie koncept wielokryterialnych algorytmów genetycznych oraz opisane zostaną dwa algorytmy wykorzystane do rozwiązania przedstawionego wcześniej problemu: NSGA-II oraz SPEA.

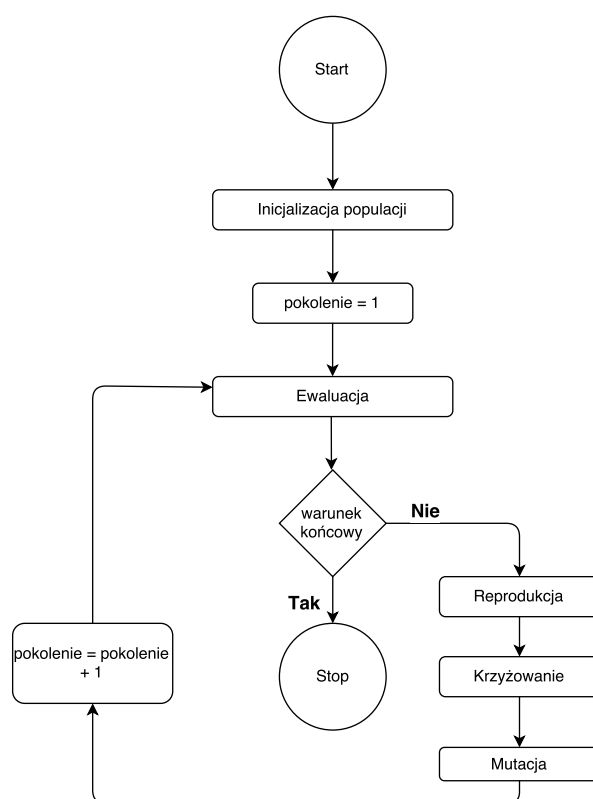
4.1. Opis ogólny

Algorytmy genetyczne są procedurami wykorzystywanymi do rozwiązywania problemów optymalizacyjnych, opartymi o genetykę oraz selekcję naturalną. Cały proces działania takiego algorytmu można porównywać do procesu ewolucji występującego w przyrodzie. Obserwując taki proces można zauważyć, że najlepiej przystosowane osobniki mają największe szanse na reprodukcję, a więc przekazanie swojego DNA kolejnym pokoleniom. Dzieje się tak dlatego, że z punktu widzenia biologii najlepiej przystosowane osobniki to te najbardziej atrakcyjne dla potencjalnych partnerów oraz w przypadku zwierząt stadnych, te dominujące całą resztę, a więc mające pierwszeństwo jeśli chodzi o wybór partnerów, reprodukcję czy chociażby pożywanie się. Generalizując natura dąży do tego, aby do puli z której utworzone ma zostać kolejne pokolenie danej populacji trafiały jak najlepsze geny. Gdyby działało się inaczej żaden gatunek nie przetrwałby więcej niż kilkudziesięciu pokoleń.

Jak zostało wspomniane największe szanse na potomstwo mają osobniki najlepsze patrząc z perspektywy szansy na przetrwanie. Potomstwo par takich osobników powstaje na skutek procesu nazywanego krzyżowaniem. Krzyżowanie polega na łączeniu genów jednego osobnika z drugim, tworząc w ten sposób potomka, charakteryzującego się cechami zarówno pierwszego jak i drugiego rodzica. Oczywiście im lepiej rodzice danego osobnika byli przystosowani, tym większa szansa, że również i on będzie stał wysoko w hierarchii.

Kolejnym istotnym elementem w procesie ewolucji jest mutacja. Mutacja jest losowym procesem polegającym na zmianie fragmentu kodu DNA. Poprzez taką zmianę osobnik staje się jeszcze lepiej lub gorzej przystosowany. W pierwszym przypadku jego szanse na reprodukcję wzrastają, więc jest większe prawdopodobieństwo, że zmiana wprowadzona przez mutację zostanie przekazana dalej. W drugim przypadku linia osobników ze zmienionym DNA prawdopodobnie wygaśnie zdecydowanie szybciej niż w przypadku pierwszym.

Podsumowując, proces ewolucji złożony jest z trzech elementów: selekcji, krzyżowania oraz mutacji. Na tych samych operacjach opiera się działanie algorytmu genetycznego. W kontekście algorytmu są one nazywane operatorami genetycznymi. Na rysunku (4.1) przedstawiony został ogólny schemat działania takiego algorytmu.



Rys. 4.1: Schemat działania algorytmu genetycznego

Punktem startowym jest utworzenie losowej populacji osobników. Następnie każdy osobnik jest oceniany przez pryzmat ustalonego kryterium. Na podstawie oceny przeprowadzany jest proces selekcji, czyli wyboru najlepszych osobników, które wykorzystane zostaną do reprodukcji. Podczas procesu reprodukcji generowane są nowe osobniki tworzące kolejne pokolenie. Każdy nowy osobnik jest wynikiem krzyżowania, czyli łączenia chromosomu dwóch rodziców w jeden. Chromosom z punktu widzenia biologii jest formą organizacji materiału genetycznego. Z punktu widzenia algorytmu będzie to zestaw parametrów charakteryzujących danego osobnika, który zostanie opisany później. Następnie chromosomy nowych osobników mogą ulec mutacji, czyli losowej zmianie pojedynczych genów. Gdy zostanie utworzona odpowiednia liczba nowych osobników, czyli powstanie nowe pokolenie, cały proces zaczyna się od początku, dopóki nie

osiągnięte zostanie zdefiniowane kryterium końcowe.

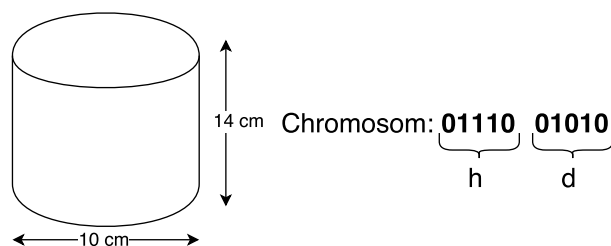
W powyższym opisie pojawiło się kilka pojęć wartych wyjaśnienia.

Populacja

Populacja jest grupą o ustalonym rozmiarze. Elementami grupy są pojedyncze osobniki. Jedną z przewag algorytmów genetycznych nad tradycyjnymi technikami optymalizacyjnymi jest właśnie operowanie nie na jednym konkretnym rozwiązaniu, lecz na całej grupie. Duża różnorodność osobników w połączeniu z odpowiednim użyciem operatorów genetycznych pozwala uniknąć problemu utknięcia na optimum lokalnym.

Chromosom

Chromosom, jak zostało już wspomniane, jest sposobem reprezentacji właściwości cechujących danego osobnika. Chromosom można zdefiniować na dwa sposoby: za pomocą ciągu binarnego lub wartości rzeczywistych. Ciąg binarny jest ciągiem wartości 0 lub 1 ułożonych w taki sposób aby odpowiednio zakodować przekazywaną informację. W ramach przykładu założmy, że mamy dany cylinder o średnicy $d = 10\text{cm}$ oraz wysokości $h = 14\text{cm}$. Rysunek (4.2) pokazuje przykładowy sposób reprezentacji takiego cylindra w formie ciągu binarnego.



Rys. 4.2: Przykład ciągu binarnego

Chromosom reprezentujący cylinder złożony jest z 10 bitów. Na pierwszych pięciu z nich zakodowana jest informacja o wysokości, a na kolejnych informacja o średnicy. Wartości przedstawione przez poszczególne części ciągu należy traktować jak liczby binarne i w ten sam sposób je interpretować.

W większości przypadków preferowane jest używanie ciągu binarnego jako sposobu reprezentacji osobników. Jednym z powodów jest łatwość przeprowadzania operacji krzyżowania oraz mutacji, o których mowa będzie później. Używając reprezentacji binarnej można w prosty sposób zmienić sposób kodowania osobników bez konieczności ingerowania w operatory genetyczne - operatory krzyżowania oraz mutacji nie biorą pod uwagę sposobu w jaki informacja została zapisana w chromosomie. Kolejne argumenty przemawiające za użyciem takiej reprezentacji przedstawione zostały w pracach Hollanda (1991)REF oraz Goldberga (1993)REF. Kolejne etapy działania algorytmów genetycznych wyjaśnione zostaną przy założeniu, że osobniki przedstawione są za pomocą ciągów binarnych.

Ewaluacja

Po ustaleniu sposobu reprezentacji osobników należy wygenerować losową populację oraz poddać ją ocenie, czyli ewaluacji. Proces ewaluacji polega na przypisaniu każdemu

osobnikowi odpowiedniej wartości pozwalającej ocenić go na tle reszty populacji. Jest to wartość dopasowania osobnika. Korzystając z wcześniejszego przykładu założmy, że podczas procesu ewaluacji zdefiniowanemu wcześniej cylindrowi przypisywany jest koszt wyprodukowania, wyrażony następującym wzorem:

$$f(d, h) = c\left(\frac{\pi d^2}{2} + \pi dh\right) \quad (4.1)$$

, gdzie c oznacza koszt materiału na cm^2 .

Zgodnie z podanymi wcześniej danymi oraz przy założeniu, że $c = 0.01$ koszt wyprodukowania danego cylindra wynosi:

$$f(10, 14) = 6$$

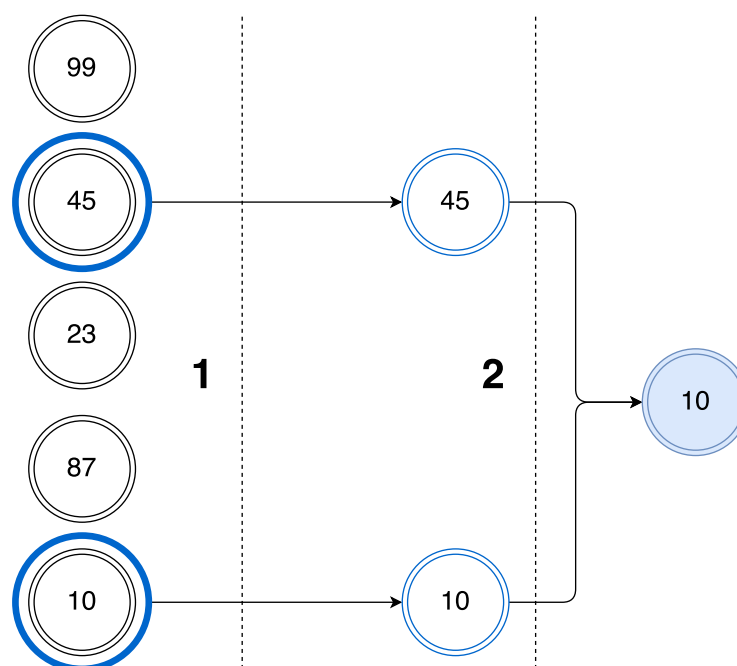
Zakładając, że celem procesu optymalizacji jest minimalizacja kosztu wyprodukowania cylindra, należy zauważyć, że osobnik z przypisaną mniejszą wartością jest lepszy w kontekście wybranego kryterium.

Reprodukcja

Kolejnym etapem działania algorytmu genetycznego jest reprodukcja inaczej zwana też selekcją. Jest to proces mający na celu zwiększenie liczby dobrych osobników w populacji oraz redukcję liczby tych gorszych, jednocześnie utrzymując niezmienny rozmiar całej populacji. Jest to osiągane poprzez realizację następujących zadań:

1. Zidentyfikowanie dobrych (ocenionych powyżej średniej) osobników.
2. Utworzenie kopii dobrych osobników.
3. Eliminacja osobników gorszych, tak aby kopie dobrych mogły zostać dodane do populacji.

Powyższe zadania mogą zostać osiągnięte przy użyciu kilku metod. Najpopularniejszymi z nich są: metoda turniejowa, metoda ruletki oraz metoda rankingu. Metoda turniejowa polega na rozgrywaniu turniejów pomiędzy kilkoma osobnikami wybranymi losowo z populacji. Zwycięzca każdego turnieju, jako rodzic, przechodzi do etapu krzyżowania, który zostanie opisany później (4.3). Rozmiar pojedynczego turnieju znacząco wpływa na przebieg procesu selekcji. Im jest większy, tym mniejsza szansa, że słabsze osobniki przejdą do kolejnego etapu. Wbrew pozorom takie zachowanie nie jest pożądane ze względu na potrzebę zachowania różnorodności populacji. Im rozmiar jest większy tym większa szansa, że do kolejnego etapu dostanie się zbyt dużo słabszych osobników, co również nie jest dobrym rozwiązaniem. Jeśli rozmiar turnieju $S = 1$ selekcja staje się równoważna losowemu wyborowi osobnika. Najczęściej stosuje się turnieje o rozmiarze 2. Metoda turniejowa jest jedną z najczęściej wybieranych metod selekcji. Jest prosta w implementacji oraz pozwala na łatwe manipulowanie rozmiarem turnieju, co z kolei wpływa na szybkość zbliżania się znalezionych rozwiązań do rozwiązania optymalnego. Ponadto jak zostało udowodnione selekcja turniejowa zawsze zapewnia lepszą lub taką samą szybkość zbliżania się do rozwiązań optymalnych jak inne metody selekcji oraz mniejszą lub taką samą złożoność obliczeniową. Z wymienionych powyżej powodów to właśnie metoda turniejowa została zaimplementowana w wybranych przez autora algorytmach.



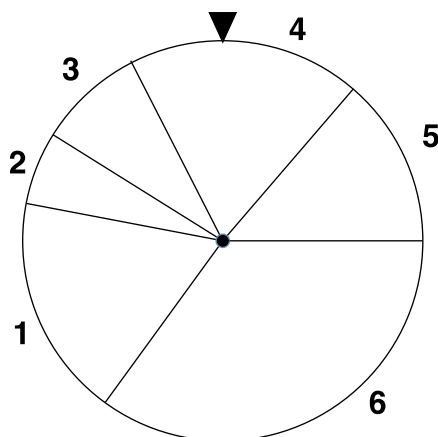
Rys. 4.3: Przykład rozegranego turnieju o rozmiarze 2.

1. Wybór losowych osobników; 2. Turniej pomiędzy wybranymi osobnikami.

Kolejną popularną metodą selekcji jest metoda proporcjonalna. W metodzie tej wprowadzone zostaje pojęcie puli rozrodczej. Jest to pula spośród której wybierane są osobniki wykorzystywane później do krzyżowania. Wypełnienie puli rozrodczej kopiami osobników wykonywane jest w procesie selekcji. Liczba kopii danego osobnika w puli jest proporcjonalna do jego wartości dopasowania. Jeśli średnia wartość dopasowania dla całej populacji wynosi f_{avg} , dany osobnik s_i będzie posiadał f_i/f_{avg} kopii w puli rozrodczej. Metoda proporcjonalna często nazywana jest również metodą ruletki, ponieważ właśnie za pomocą koła ruletki można zobrazować sobie jej działanie. Wspomniane koło jest podzielone na N części, gdzie N jest wielkością populacji. Każdy wycinek koła odpowiada jednemu osobnikowi z populacji. Wielkość danego wycinka jest proporcjonalna do wartości dopasowania osobnika. Wynika z tego, że wycinki powiązane z lepszymi osobnikami będą zajmować większą część koła. Jak można zauważyć na rysunku (4.4) osobnik o numerze 6 jest najlepszy na tle całej populacji, natomiast osobnik o numerze 2 jest osobnikiem najgorszym. Wybór osobników do puli rozrodczej odbywa się poprzez kręcenie kołem. Czynność ta powtarzana jest N razy. Po zatrzymaniu wybierany jest osobnik na którego wycinku zatrzymał się wskaźnik. Osobniki z najlepszą wartością dopasowania mają największe szanse na trafienie do puli rozrodczej.

Wadami metody proporcjonalnej jest duża złożoność obliczeniowa oraz problemy ze skalowaniem. Załóżmy, że jeden osobnik jest znacząco lepszy od reszty. Wtedy powiązany z nim wycinek będzie zajmował niemal całe koło, a co za tym idzie prawdopodobieństwo wyboru takiego osobnika będzie zbliżone do 1. W konsekwencji pula rozrodcza będzie w większości zapełniona kopiami jednego osobnika, co prowadzi do braku zróżnicowania. Z drugiej strony jeśli wszystkie osobniki mają bardzo zbliżone wartości dopasowania, każdy z nich ma również podobne prawdopodobieństwo wyboru do puli. Prowadzi to do sytuacji, kiedy każdy osobnik ma dokładnie jedną kopię w

puli rozrodczej. Taki proces selekcji nie wnosi żadnej wartości i równie dobrze mógłby zostać pominięty.

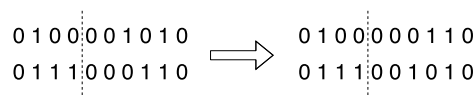


Rys. 4.4: Przykładowa wizualizacja selekcji proporcjonalnej

Opisanych powyżej problemów ze skalowaniem można uniknąć korzystając z metody rankingowej. Polega ona na posortowaniu osobników z populacji ze względu na ich wartość dopasowania. W ten sposób najgorszemu osobnikowi zostaje przypisane pierwsze miejsce w tworzonym rankingu. Osobnik najlepszy znajdzie się wtedy na N -tym miejscu. Po takim przygotowaniu na posortowanej populacji zostaje użyty operator selekcji proporcjonalnej. Na tym etapie przygotowanie koła ruletki nie opiera się już na wartości dopasowania poszczególnych osobników, lecz na ich pozycji w rankingu. Pozwala to na pozbycie się opisanych wcześniej problemów.

Krzyżowanie

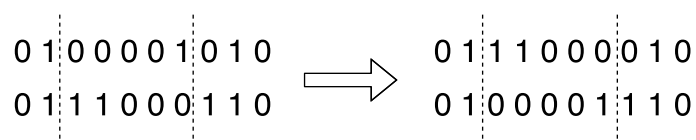
Po wyborze osobników mających zostać rodzicami następuje czas na skorzystanie z operatora krzyżowania. Tutaj należy zwrócić uwagę na to, że opisany wcześniej proces reprodukcji nie tworzy żadnych nowych osobników. Dzieje się to dopiero w trakcie procesu krzyżowania. Jak wskazuje nazwa, krzyżowanie polega na łączeniu dwóch osobników w celu utworzenia ich potomstwa. Dzieje się to na poziomie zdefiniowanych wcześniej chromosomów. Najprostszą i najlepiej ilustrującą ideę krzyżowania metodą jest metoda jednopunktowa. Polega ona na losowym wybraniu punktu względem którego nastąpi podział chromosomu na dwie części. Potomstwo powstaje na skutek wymiany pomiędzy rodzicami części chromosomu znajdującej się po prawej stronie od wybranego punktu podziału. Proces ten został przedstawiony na rysunku (4.5). Po lewej stronie znajdują się chromosomy pary rodziców. Punkt podziału został losowo ustalony na $d = 4$, co symbolizuje pionowa przerywana linia. Po prawej stronie znajduje się potomstwo - wynik zastosowania operatora krzyżowania. Jak widać chromosom każdego nowego osobnika powstał na skutek złożenia chromosomów rodziców.



Rys. 4.5: Przykład krzyżowania jednopunktowego

W tym miejscu należy zauważyć, że nowe osobniki utworzone w procesie krzyżowania niekoniecznie muszą być lepsze od swoich rodziców. Może zdarzyć się tak, że potomstwo będzie gorzej przystosowane. Nie należy się tym jednak martwić, ponieważ jak zostało wyjaśnione wcześniej osobniki o mniejszej wartości przystosowania mają mniejsze szanse na przetrwanie selekcji, a co za tym idzie ich geny nie zostaną przekazane kolejnym pokoleniom. Ponadto jest znacznie bardziej prawdopodobne, że operatory krzyżowania wygenerują rozwiązania lepsze niż generowanie losowe. Dzieje się tak dlatego, że chromosomy wybrane do procesu nie są przypadkowe. Przeszły one selekcję co wskazuje na to, że zawierają w sobie wartościowe kombinacje bitów. Operator krzyżowania użyty na parze rodziców, może wygenerować jedynie l par potomków, gdzie l jest długością chromosomu. Biorąc pod uwagę fakt, że potomstwo jest generowane przy użyciu chromosomów dobrych osobników, istnieje duże prawdopodobieństwo, że utworzone rozwiązania również będą dobre.

Opisany powyżej proces krzyżowania może zostać przeprowadzony również przy użyciu kilku punktów podziału. Na rysunku (4.6) przedstawiona jest selekcja dwupunktowa. Chromosom dzielony jest na trzy części. Potomstwo powstaje na skutek wymiany części środkowej.



Rys. 4.6: Przykład krzyżowania dwupunktowego

Przedstawioną ideę podziału chromosomu można dowolnie rozszerzać. Ekstremum jest tzw. podział jednolity. W takim przypadku potomstwo powstaje poprzez wymianę pojedynczych bitów z określonym prawdopodobieństwem p . Najczęściej przyjmuje się, że $p = 0.5$, czyli istnieje 50% szansy, że dany bit zostanie wymieniony na ten pochodzący z drugiego rodzica.

W wybranych algorytmach autor stosuje metodę krzyżowania dwupunktowego, która jest typowym wyborem przy chromosomach reprezentowanych za pomocą ciągów binarnych.

W celu zachowania części dobrych osobników z populacji, do procesu krzyżowania dodaje się dodatkową zmienną p_c , oznaczającą prawdopodobieństwo krzyżowania. Oznacza ona jaką część obecnej populacji zostanie użyta do procesu krzyżowania. Gdy $p_c = 1$ nowa populacja w całości zostanie zapełniona potomkami osobników ze starej populacji. W innym przypadku $100(1 - p_c)\%$ osobników ze starej populacji zostanie skopiowanych do nowej. Najczęściej prawdopodobieństwo krzyżowania ustawia się na poziomie 80-90%.

Mutacja

Nowe osobniki powstałe na skutek krzyżowania mogą ulec mutacji. Mutacja jest odpowiedzialna za losową modyfikację chromosomu danego osobnika. Operator mutacji zapewnia utrzymanie różnorodności populacji na odpowiednim poziomie. Działanie tego operatora opiera się na zmiennej wyrażającej prawdopodobieństwo mutacji: p_m . Operator nie działa na poziomie całego chromosomu, lecz pojedynczych bitów, a jego

zastosowanie jest równoznaczne ze zmianą wartości danego bitu. Zostało to zilustrowane na rysunku (4.7).

$$01\boxed{0000}1010 \Rightarrow 01\boxed{1}0001010$$

Rys. 4.7: Przykład mutacji

Podobnie jak w przypadku zastosowania operatora krzyżowania tak i w przypadku mutacji, osobnik będący rezultatem może okazać się gorszy od oryginału. Mimo to, warto zauważyć, że używanie operatora mutacji przy użyciu niskiego prawdopodobieństwa p_m nie jest operacją losową. Powodem jest to, że rozwiązanie generowane jest na podstawie już istniejącego osobnika oraz cały proces może utworzyć jedynie kilka nowych rozwiązań, z dużym prawdopodobieństwem, również dobrych.

Podsumowując, algorytmy genetyczne używają trzech operatorów: selekcji, krzyżowania oraz mutacji. Operator selekcji odpowiedzialny jest za wybór najlepiej przystosowanych osobników. Operator krzyżowania służy do łączenia wybranych wcześniej osobników w celu utworzenia potomstwa. Operator mutacji losowo zmienia część chromosomu danego osobnika w nadziei na znalezienie jeszcze lepszego rozwiązania. Należy tutaj zwrócić uwagę, że żaden z opisanych operatorów nie jest deterministyczny. Każdy z nich oparty jest w znacznym stopniu o losowość, co powoduje, że nie zawsze będą one osiągać zamierzone rezultaty. Nie mniej jednak w kontekście działania algorytmu wymagane jest, aby gorsze osobniki były eliminowane w procesie reprodukcji, na rzecz osobników lepszych, których geny powinny być przekazywane do kolejnych pokoleń.

4.2. Algorytmy wielokryterialne

Główną różnicą pomiędzy klasycznymi metodami optymalizacji, a algorytmami genetycznymi jest fakt, że te drugie nie operują na jednym rozwiązaniu ale na całej populacji rozwiązań. Różnica ta daje ogromną przewagę algorytmom genetycznym jeśli chodzi o rozwiązywanie wielokryterialnych problemów optymalizacyjnych. Jak zostało wspomniane w rozdziale ?? celem optymalizacji wielokryterialnej jest znalezienie jak największej liczby rozwiązań optymalnych w sensie Pareto. Algorytmy genetyczne dają możliwość znalezienia grupy takich rozwiązań jako rezultatu uruchomienia jednej symulacji. Eliminuje to konieczność wielokrotnego używania metod optymalizacji jednokryterialnej w celu znajdowania kolejnych optymalnych rozwiązań. Ponadto używając algorytmów genetycznych można pozbyć się wielu parametrów koniecznych przy używaniu metod klasycznych. Przykładem może być chociażby wektor wag, wykorzystywany do transformacji problemu wielokryterialnego w problem jednokryterialny. Jest on konieczny, ponieważ konkretne ustawienie takiego wektora jest ściśle powiązane z jednym znalezionym rozwiązaniem optymalnym. Jeśli chcemy znaleźć inne rozwiązanie należy zmienić ustawienia wektora oraz ponownie uruchomić symulację. W przypadku algorytmów genetycznych nie będzie to konieczne, ponieważ jak zostało wspomniane rezultatem zwróconym przez taki algorytm będzie nie pojedyncze rozwiązanie ale cała grupa rozwiązań.

Co więcej operatory genetyczne mogą zostać zmodyfikowane tak, aby faworyzować rozwiązania niezdominowane oraz jednocześnie utrzymywać różnorodność populacji na odpowiednim poziomie. Te dwie cechy połączone z ogólną ideą algorytmów genetycznych prowadzą do znalezienia zróżnicowanej grupy osobników optymalnych w sensie Pareto, czyli rozwiązania wielokryterialnego problemu optymalizacyjnego.

W kolejnych podrozdziałach opisane zostaną dwa algorytmy spełniające powyższe założenia: NSGA-II oraz SPEA. Wyjaśnione zostanie na jakich zasadach działają oba algorytmy oraz w jaki sposób modyfikują poszczególne operatory genetyczne.

4.2.1. NSGA-II

NSGA-II jest drugą wersją zaproponowanego w 2000 roku algorytmu NSGA (Elist Non-Dominated Sorting Genetic Algorithm). Jego działanie opiera się na dwóch koncepcjach: elitaryzmie oraz strategii zachowania różnorodności w populacji.

Elitaryzm

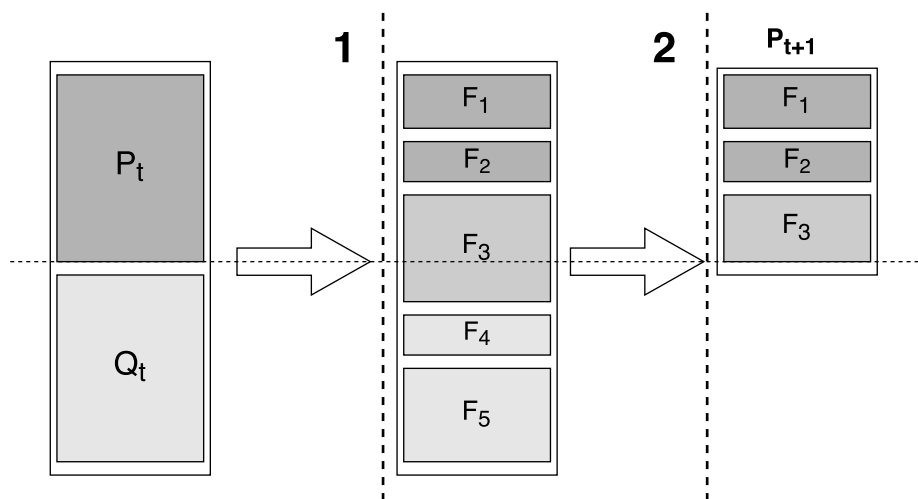
Elitaryzm jest dodatkowym operatorem używanym w algorytmach genetycznych. Jego rolą jest zwiększenie szansy, że najlepsze rozwiązania z danego pokolenia (elita) zostaną bezpośrednio przeniesione do pokolenia następnego. Istnieje wiele sposobów na zdefiniowanie operatora elitaryzmu. W przypadku problemów jednokryterialnych przykładem może być porównywanie potomków utworzonych w procesie krzyżowania z ich rodzicami oraz zachowanie dwóch najlepszych osobników. W globalnego punktu widzenia elitaryzm może być wprowadzony poprzez łączenie rodziców oraz potomków w jedną populację, na której używane są później operatory genetyczne. W ten sposób rodzice również mają szansę na rywalizację ze swoim potomstwem.

Celem elitaryzmu jest zapewnienie, że wartość przystosowania najlepszego osobnika w danym pokoleniu nie jest gorsza od jego odpowiednika z poprzedniego pokolenia. Operator elitaryzmu, że najlepsze rozwiązania znalezione na początku symulacji nie zostaną zgubione. Ponadto ich obecność w kolejnych populacjach zwiększa szansę na utworzenie równie dobrego potomstwa.

Podsumowując, odpowiednio użyty elitaryzm zapewnia lepsze działanie algorytmów genetycznych. Należy jednak zwrócić uwagę na to jaki procent najlepszych osobników można nazwać elitą. Jest to określane przez zmienną α . Jeśli procent ten jest duży, prawdopodobne jest, że w następnych pokoleniach populacja zacznie skupiać się wokół najlepszych osobników, prowadząc do zmniejszenia różnorodności. Jeśli procent będzie za mały wpływ elitaryzmu może zostać nadmiernie zredukowany nie wprowadzając żadnego pozytywnego wpływu. Wartości pomiędzy $\alpha = 1$, a $\alpha = 0.1N$, gdzie N jest wielkością populacji, są najbardziej popularne.

W NSGA-II początkowym krokiem jest połączenie populacji rodziców P_t z populacją potomstwa Q_t . Wskutek tego powstaje grupa R_t o rozmiarze $2N$, gdzie N jest rozmiarem populacji. Jest to grupa, na której będzie pracował algorytm. Następnym krokiem jest poddanie R_t sortowaniu rozwiązań niezdominowanych, które zostało opisane w rozdziale ???. Następnie tworzona zostaje nowa populacja P_{t+1} o rozmiarze N . Dzieje się to poprzez wypełnianie nowej populacji osobnikami należącymi do danego frontu grupy R_t , zaczynając od frontu najlepszego, czyli pierwszego. Proces ten jest kontynuowany aż do momentu kiedy w nowej populacji nie ma już miejsca na kolejne

osobniki. Warto zwrócić uwagę, że osobniki przenoszone są z grupy o rozmiarze $2N$ do grupy o rozmiarze N . Przez to nie wszystkie fronty mogą zostać przeniesione. Fronty, które nie mieszczą się w nowej populacji są po prostu usuwane. Inaczej sytuacja wygląda, gdy dany front może zmieścić się tylko częściowo. Zostało to przedstawione na rysunku (4.8). Wtedy na danym froncie stosowane jest sortowanie oparte o lokalną wartość zagęszczenia, która zdefiniowana zostanie później. Następnie do nowej populacji zostaje przeniesionych n brakujących osobników, które podczas sortowania zostały uznane za najlepsze.



Rys. 4.8: Uproszczony schemat działania algorytmu NSGA-II.

1. Sortowanie rozwiązań niezdominowanych.
2. Sortowanie w oparciu o lokalną wartość zagęszczenia.

Na nowej populacji P_{t+1} stosowany jest następnie zmodyfikowany operator selekcji. Użycie takiego operatora zakłada, że każdy osobnik charakteryzuje się dwoma cechami:

1. Numerem frontu, na którym się znajduje.
2. Lokalną wartością zagęszczenia, która opisana zostanie później.

Bazując na tych dwóch cechach, osobnik i jest lepszy od osobnika j , jeśli:

1. Osobnik i znajduje się na lepszym froncie niż osobnik j .
2. Osobnik i znajduje się na tym samym froncie co osobnik j ale charakteryzuje się mniejszą lokalną wartością zagęszczenia.

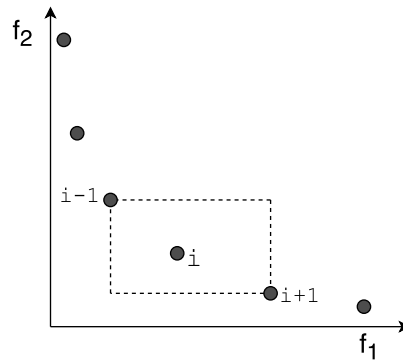
Pierwszy warunek jest odzwierciedleniem konceptu dominacji. Osobnik leżący na lepszym froncie dominuje osobnika znajdującego się na froncie gorszym. Drugi warunek został wprowadzony, aby rozstrzygać, który z osobników jest bardziej wartościowy jeśli oba leżą na tym samym froncie. Jak zostało wyjaśnione wcześniej rozwiązując wielokryterialne problemy optymalizacyjne należy skupić się nie tylko na jakości rozwiązań ale również ich różnorodności. Różnorodność danego rozwiązania na tle całości populacji jest definiowana za pomocą metryki określającej zagęszczenie opisanej w ??.

Polega ona na obliczeniu średniego dystansu pomiędzy dwoma najbliższymi sąsiadami danego osobnika, znajdującymi się po jego przeciwnych stronach. Zostało to zilustrowane na rysunku (4.9). Takie obliczenia są przeprowadzane m razy dla każdego osobnika, gdzie m jest liczbą zdefiniowanych kryteriów. Następnie wyniki otrzymane dla każdego kryterium są agregowane do jednej wartości d_i . Generalnie algorytm obliczający zagęszczenie populacji można przedstawić następująco:

1. Dla każdego osobnika z populacji przypisz $d_i = 0$ oraz zdefiniuj l jako liczbę osobników.
2. Dla każdego kryterium m , posortuj populację względem wartości funkcji oceny danego kryterium f_m , tworząc wektor I^m
3. Pierwszemu i ostatniemu osobnikowi z wektora I^m przypisz dużą wartość zagęszczenia: $d_{I_1^m} = d_{I_l^m} = \infty$. Reszcie osobników przypisz wartość zagęszczenia korzystając ze wzoru:

$$d_{I_j^m} = d_{I_j^m} + \frac{f_m^{(I_{j+1}^m)} - f_m^{(I_{j-1}^m)}}{f_m^{max} - f_m^{min}} \quad (4.2)$$

I_j oznacza osobnika znajdującego się na j -ej pozycji w posortowanym wektorze I^m . Wynika z tego, że I_1 oraz I_l są osobnikami charakteryzującymi się kolejno najgorszą oraz najlepszą wartością funkcji oceny w porównaniu do całości populacji. Prawa strona równania (4.2) jest różnicą wartości funkcji oceny dwóch najbliższych sąsiadów osobnika I_j . Jak łatwo zauważyć różnica ta jest połową obwodu prostokąta przedstawionego na rysunku (4.9). Dodatkowo wyliczony dystans jest dzielony przez różnicę pomiędzy największą (f_m^{max}), a najmniejszą (f_m^{min}) wartością przyjmowaną przez funkcję oceny f_m .



Rys. 4.9: Reprezentacja najbliższego sąsiedztwa punktu i

Podsumowując schemat działania algorytmu NSGA-II wygląda następująco:

1. Połącz populację rodziców P_t z populacją potomków Q_t tworząc grupę R_t .
2. Na R_t zastosuj sortowanie rozwiązań niezdominowanych w celu zdefiniowania frontów rozwiązań.
3. Utwórz nową populację P_{t+1} za pomocą opisanej wcześniej procedury.

4. Utwórz nową populację potomków Q_{t+1} za pomocą zmodyfikowanego operatora selekcji oraz standardowych operatorów krzyżowania oraz mutacji.

W opisanym powyżej algorytmie NSGA-II różnorodność znalezionych rozwiązań niezdominowanych jest zapewniona poprzez zastosowanie zmodyfikowanego operatora selekcji. Osobniki rywalizują ze sobą opierając swoją wartość o front, na którym się znajdują oraz lokalną wartość zagęszczenia. Wartość zagęszczenia staje się szczególnie ważna w późniejszych momentach symulacji, kiedy większość osobników znajduje się już na najlepszym froncie. Biorąc pod uwagę fakt, że algorytm operuje na grupie o rozmiarze $2N$ istnieje duża szansa, że liczba osobników z pierwszego frontu jest większa niż N , a więc w celu wybrania najlepszych osobników wzięta zostanie pod uwagę wartość zagęszczenia. W konsekwencji rezultatem będzie zbiór rozwiązań niezdominowanych, cechujących się wysokim stopniem różnorodności.

4.2.2. SPEA

SPEA (Strength Pareto Evolutionary Algorithm) jest algorytmem zaproponowanym w 1998 roku. Koncept elitaryzmu jest tutaj wprowadzony w postaci zewnętrznej populacji \bar{P} . W populacji tej przechowywane są wszystkie niezdominowane rozwiązania znalezione w trakcie działania algorytmu. SPEA oprócz przechowywania najlepszych rozwiązań wykorzystuje je także w procesie reprodukcji, co ma znaczący wpływ na jakość generowanych potomków.

Pierwszym krokiem algorytmu jest utworzenie losowej populacji P_0 oraz pustej zewnętrznej populacji \bar{P}_0 o ustalonym rozmiarze, nie większym niż N , gdzie N jest wielkością populacji P_0 . Na początku każdej iteracji t , na populacji P_t stosowane jest sortowanie rozwiązań niezdominowanych, w celu znalezienia najlepszego frontu. Osobniki znajdujące się na tym froncie są dodawane do populacji zewnętrznej. Następnie z populacji zewnętrznej usuwane są osobniki zdominowane przez nowo dodane rozwiązania. W ten sposób w populacji zewnętrznej zawsze będą znajdować się jedynie osobniki najlepsze. Łatwo uświadomić sobie, że wraz z działaniem algorytmu wielkość zewnętrznej populacji będzie rosła, aż do osiągnięcia zdefiniowanego wcześniej rozmiaru. Gdy ten zostanie osiągnięty pojawia się problem co robić z elitami znajdowanymi w kolejnych pokoleniach, skoro w populacji zewnętrznej nie ma już dla nich miejsca. Rozwiązaniem jest posłużenie się wartością zagęszczenia, opisaną w poprzednim podrozdziale. Do populacji zewnętrznej trafiają osobniki charakteryzujące się największą różnorodnością na tle populacji. Jak zostało zasugerowane ?? wartość zagęszczenia może zostać zdefiniowana przy użyciu metryki wykorzystanej w algorytmie NSGA-II.

Kiedy populacja zewnętrzna zostanie zaktualizowana, następuje proces ewaluacji. Proces ten zostaje najpierw wykonany na populacji zewnętrznej. Dla każdego osobnika i przypisywana zostaje wartość, zdefiniowana jako *siła* S_i . Siła danego osobnika z zewnętrznej populacji wyrażona jest wzorem:

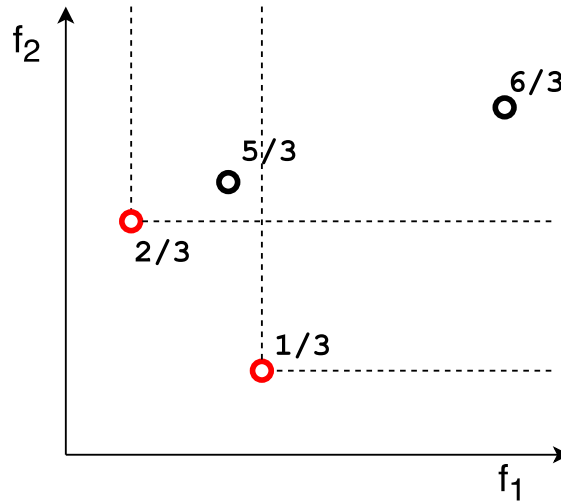
$$S_i = \frac{n_i}{N + 1} \quad (4.3)$$

, gdzie n_i określa liczbę osobników populacji, która jest zdominowana przez osobnika i . Jak łatwo zauważyć, osobnik dominujący więcej rozwiązań będzie charakteryzował się wyższą wartością funkcji S_i .

Osobniki z populacji P_i oceniane są przy użyciu następującej funkcji:

$$F_j = 1 + \sum_{i \in \overline{P}_t \wedge i \leq j} S_i \quad (4.4)$$

Liczba 1 zostaje dodana to po to, aby zapewnić, że wartość oceny każdego rozwiązania będzie większa od siły osobników znajdujących się w populacji zewnętrznej. Ponadto dodawana jest suma siły każdego osobnika, przez którego danego rozwiązanie j jest słabo zdominowane. W tym miejscu warto zauważyć, że im mniejsza jest wartość funkcji oceny F_j , tym lepszy jest dany osobnik. Przykład procesu ewaluacji został przedstawiony na rysunku (4.10). Osobniki zaznaczone na czerwone należą do populacji zewnętrznej. Jak łatwo zauważyć w procesie reprodukcji faworyzowane będą osobniki charakteryzujące się mniejszą wartością funkcji oceny, a więc te z populacji zewnętrznej.



Rys. 4.10: Przypisanie wartości funkcji oceny

Po zakończeniu procesu ewaluacji, reszta procesu przebiega według standardowego schematu. Należy jednak pamiętać, że wszystkie operatory genetyczne działają zarówno na populacji P_t jak i populacji zewnętrznej \overline{P}_t .

Podsumowując schemat działania algorytmu SPEA wygląda następująco:

1. Zastosuj sortowanie niezdominowanych rozwiązań na populacji P_t w celu znalezienia rozwiązań niezdominowanych.
2. Skopiuj znalezione rozwiązania do populacji zewnętrznej \overline{P}_t .
3. Z populacji zewnętrznej usuń rozwiązania zdominowane.
4. Jeśli rozmiar populacji zewnętrznej jest większy niż N , zastosuj sortowanie w oparciu o wartość zagęszczenia oraz zredukuj rozmiar populacji zewnętrznej do N pozbywając się najgorszych osobników.
5. Oceń wszystkie rozwiązania korzystając z opisanych wzorów (4.3, 4.4).
6. Utwórz nową populację korzystając z operatorów selekcji, krzyżowania oraz mutacji.

Analizując przedstawiony algorytm, łatwo zauważyć, że raz znalezione rozwiązanie optymalne w sensie Pareto nigdy nie przepadnie, ponieważ będzie przechowywane w populacji zewnętrznej. Jedynym sposobem na pozbycie się takiego rozwiązania jest znalezienie kolejnego rozwiązania optymalnego w sensie Pareto, lecz charakteryzującego się lepszą wartością zagęszczenia. Takie podejście zapewnia jakość oraz różnorodność znajdujących osobników.

Duże znaczenie dla działania algorytmu ma wielkość zewnętrznej populacji \bar{N} . Jeśli ta jest duża (porównywalna do N) nacisk na selekcję rozwiązań z populacji zewnętrznej będzie większy, co może prowadzić do tego, że algorytm nie będzie potrafił wygenerować rozwiązań leżących na prawdziwym froncie Pareto. Gdy \bar{N} będzie zbyt małe, wpływ elitaryzmu będzie niezauważalny. Ponadto może to prowadzić do sytuacji w której wiele rozwiązań z populacji nie będzie zdominowanej przez żadnego osobnika z populacji zewnętrznej. Będzie skutkować to tym, że wartość funkcji oceny dla tych osobników będzie tak sama. Proponowaną wielkością populacji zewnętrznej jest $1/4$ wielkości populacji N .

4.3. Strojenie

Rozdział 5

Systemy wspomaganie decyzji

Rozdział 6

Rozwiązanie problemu

6.1. System wspomagania decyzji

6.1.1. Architektura

6.1.2. Interakcja z użytkownikiem

6.2. Optymalizacja

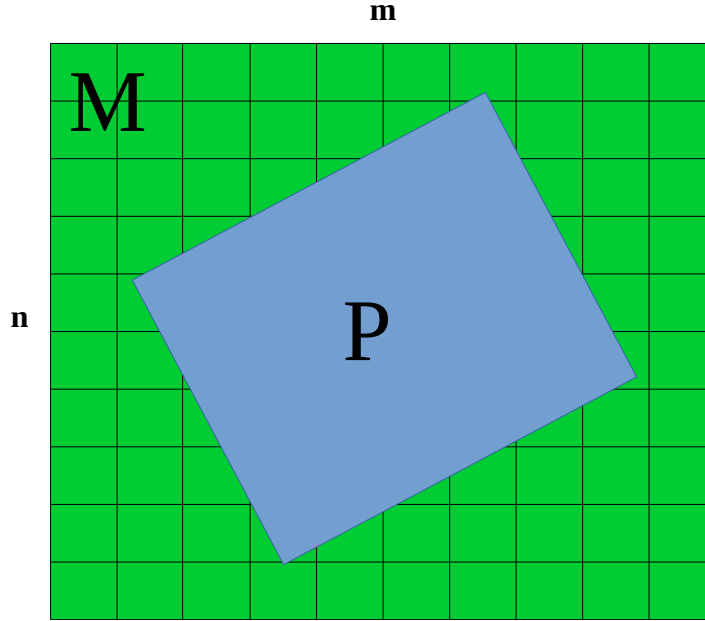
6.2.1. Model matematyczny

Założmy, że P jest wielokątem odwzorowującym kształt terenu, który ma zostać nawodniony. Poprzez opisanie danego wielokąta prostokątem otrzymujemy teren roboczy A . Szerokość i wysokość tego terenu są dodatkowo powiększone o największy możliwy zasięg zraszaczy, tak aby umożliwić odpowiednie sprawdzenie stopnia naruszenia ograniczeń, o których mowa później. A jest dodatkowo podzielony na $m \times n$ kwadratów tworząc macierz M przedstawioną na rysunku (6.1). Każdy element macierzy może przyjmować wartości od 0 do ms , gdzie ms jest maksymalną liczbą zraszaczy. Wartość 0 oznacza, że dany kwadrat nie został podlany. Wartość większa od 0 mówi o tym przez ile zraszaczy dany obszar został nawodniony.

Założmy, że $p(x, y)$ jest funkcją mówiącą o tym czy punkt o współrzędnych (x, y) znajduje się wewnątrz P . Jeśli $p = 0$ punkt należy do P , w przeciwnym wypadku $p = 1$. Zraszacz jest definiowany jako okrąg reprezentowany przez promień r_i , współrzędne środka (x_i, y_i) oraz konkretny model t_i . Środek każdego zraszacza musi zawierać się w P , czyli $p(x_i, y_i) = 0$.

Kwadrat o współrzędnych (x, y) jest nawodniony jeśli znajduje się w zasięgu zraszacza. Jest to wyrażone następującą funkcją:

$$is_irr(x, y, s_i) = \begin{cases} 1, & \text{jeśli } (x - x_i)^2 + (y - y_i)^2 \leq r_i^2 \\ 0, & w.p.p \end{cases} \quad (6.1)$$



Rys. 6.1: blah

, gdzie x oraz y oznaczają współrzędne środka kwadratu; s_i jest konkretnym zraszaczem; x_i, y_i oznaczają współrzędne zraszacza s_i ; r_i jest liczbą wyrażającą zasięg zraszacza.

Założmy, że S jest zbiorem wszystkich rozmieszczonych zraszaczy. Wtedy dany kwadrat jest nawodniony jeśli znajduje się w zasięgu przynajmniej jednego zraszacza ze zbioru S :

$$irr(x, y, S) = \begin{cases} 1, & \text{jeśli } \sum_{i=1}^S is_irr(x, y, s_i) > 0 \\ 0, & w.p.p \end{cases} \quad (6.2)$$

Generalnie poprzez stopień nawodnienia danego obszaru rozumie się różnicę pomiędzy liczbą wszystkich elementów macierzy M należących do P , a liczbą nawodnionych elementów tej macierzy należących do P :

$$F_0(S) = \sum_{m=1}^M \sum_{n=1}^N p(m, n) \cdot (1 - irr(m, n, S)) \quad (6.3)$$

Im wartość funkcji F_0 jest mniejsza tym mniej jest obszarów nienawodnionych, a więc rozwiązanie jest lepsze. $F_0 =$ jest rozwiązaniem idealnym z perspektywy kryterium nawodnienia.

Jak zostało wspomniane w poprzednich rozdziałach zaimplementowane algorytmy oceniają stopień nawodnienia obszaru na dwa sposoby w zależności od tego czy wybrana jest opcja minimalizacji nawodnienia poza wskazanym obszarem. Jeśli minimalizacja jest wyłączona do obliczeń używana jest prosta funkcja opisana powyżej (6.3). Bardziej

szczegółowe rozwinięcie funkcji przewiduje wprowadzenie kary za naruszenie ograniczeń:

$$F_1(S) = \sum_{m=1}^M \sum_{n=1}^N p(m, n) \cdot (1 - irr(m, n, S)) + \alpha \cdot \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (1 - p(m, n)) \cdot irr(m, n, S) \cdot 100}{\sum_{m=1}^M \sum_{n=1}^N p(m, n)} + \frac{\sum_{m=1}^M \sum_{n=1}^N p(m, n) \cdot ovrirr(m, n, S) \cdot 100}{\sum_{m=1}^M \sum_{n=1}^N p(m, n)} \right] \quad (6.4)$$

, gdzie α jest zmienną określającą to czy na funkcję celu wpływa minimalizacja nawodnienia poza wskazanym obszarem. Zmienna α może przyjmować wartość 0 lub 1. Jeśli $\alpha = 1$ minimalizacja jest brana pod uwagę; $ovrirr(m, n, S)$ jest funkcją mówiącą czy dany obszar został nadmiernie nawodniony (6.5).

$$ovrirr(x, y, S) = \begin{cases} 1, & \text{jeśli } \sum_{i=1}^S is_irr(x, y, s_i) > 1 \\ 0, & \text{w.p.p} \end{cases} \quad (6.5)$$

Jak zostało wspomniane wcześniej, każdy rozmieszczony zraszacz charakteryzowany jest przez konkretny model. Załóżmy, że $C(t_i)$ jest funkcją zwracającą cenę rynkową dla danego modelu t_i . Wtedy całkowity koszt konkretnego rozwiązania wynosi:

$$F_2(S) = \sum_{i=1}^S C(t_i) \quad (6.6)$$

Podsumowując celem zadania jest rozwiązanie następującego problemu optymalizacyjnego:

$$\min F_1(S) = \sum_{m=1}^M \sum_{n=1}^N p(m, n) \cdot (1 - irr(m, n, S)) + \alpha \cdot \left[\frac{\sum_{m=1}^M \sum_{n=1}^N (1 - p(m, n)) \cdot irr(m, n, S) \cdot 100}{\sum_{m=1}^M \sum_{n=1}^N p(m, n)} + \frac{\sum_{m=1}^M \sum_{n=1}^N p(m, n) \cdot ovrirr(m, n, S) \cdot 100}{\sum_{m=1}^M \sum_{n=1}^N p(m, n)} \right] \quad (6.7)$$

$$\min F_2(S) = \sum_{i=1}^S C(t_i)$$

6.2.2. Porównanie algorytmów genetycznych

Plan badań

Złożoność oceny algorytmów służących do optymalizacji wielokryterialnej jest znacznie większa niż tej dla algorytmów rozwiązujących problemy obejmujące jedno kryterium. Wynika tak z samej definicji rozwiązania problemu wielokryterialnego, która

wyjaśniona została wcześniej. Oceniając rozwiązanie takiego problemu należy wziąć pod uwagę następujące składowe:

- Dystans znalezionego niezdominowanego zbioru rozwiązań od prawdziwego zbioru Pareto
- Równomierny rozkład znalezionych rozwiązań
- Długość przedziału pokrytego przez znalezione rozwiązania
- Czas wykonania algorytmu

Na potrzeby analizy i oceny dwóch zaproponowanych algorytmów wybrane zostały cztery metryki:

1. *Współczynnik błędu (ER)*: wskazuje procent rozwiązań, które nie należą do prawdziwego zbioru Pareto.

$$ER = \frac{\sum_{i=1}^n e_i}{n} \quad (6.8)$$

Gdzie n jest liczbą znalezionych niezdominowanych rozwiązań; e_i zmienną określającą czy dane rozwiązanie znajduje się w prawdziwym zbiorze Pareto. Jeśli $e_i = 0$ rozwiązanie znajduje się w prawdziwym zbiorze Pareto, w przeciwnym razie $e_i = 1$. $ER = 0$ oznacza idealne rozwiązanie.

2. *Równomierny rozkład (SP)*: metoda mierząca wariancję dystansu sąsiadujących rozwiązań znajdujących się na znalezionym froncie Pareto.

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (6.9)$$

Gdzie n jest liczbą znalezionych niezdominowanych rozwiązań; d_i oznacza dystans Euklidesowy pomiędzy danym rozwiązaniem, a jego najbliższym sąsiadem; \bar{d} jest wartością oznaczającą średni dystans pomiędzy rozwiązaniami. Wartość $SP = 0$ oznacza, że znalezione rozwiązania są rozłożone równomiernie.

3. *Jakość ogólna (DG)*: wskazuje jak daleko od prawdziwego zbioru Pareto znajduje się zbiór znalezionych niezdominowanych rozwiązań. Miara zdefiniowana w następujący sposób:

$$DG = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (6.10)$$

Gdzie n oznacza liczbę znalezionych niezdominowanych rozwiązań; d_i jest dystansem Euklidesowym pomiędzy danym rozwiązaniem, a najbliższym rozwiązaniem należącym do prawdziwego frontu Pareto. Wartość $DG = 0$ oznacza, że wszystkie znalezione rozwiązania leżą na prawdziwym froncie Pareto.

4. *Długość frontu (FE)*: metoda wskazująca jak duży obszar jest pokryty przez znalezione niezdominowane rozwiązania.

$$FE = \sum_{k=1}^K \max_{i,j:l=i} \sqrt{(f_i^k - f_j^k)^2} \quad (6.11)$$

Tabela 6.1: Przypadki testowe

Wielkość	Obszar
Mały	1
Średni	2
Duży	3

Gdzie K oznacza liczbę funkcji celu; f_i^k jest wartością i -tego rozwiązania z perspektywy kryterium k .

Aby skorzystać z powyższych metryk konieczne jest znalezienie prawdziwego frontu Pareto. Na potrzeby badań został on wygenerowany poprzez wybranie niezdominowanych rozwiązań ze skumulowanej puli rozwiązań powstałej na skutek uruchomienia symulacji 10 razy dla każdego algorytmu. Rezultaty każdej symulacji były dodawane do puli po czym zredukowane do rozwiązań niezdominowanych.

Dla badań przygotowanych zostało 6 przypadków testowych. Zostały one podzielone ze względu na wielkość obszaru mającego zostać nawodniony oraz tego czy algorytm ma minimalizować nawadnianie poza zaznaczonym obszarem. Kształt wybranych obszarów został dobrany tak, aby jak najlepiej wizualizować wpływ ograniczeń na działanie algorytmów.

Rezultaty badań

Podsumowanie badań

Rozdział 7

Podsumowanie

Definicja 1

Definicja - pierwsza

Dodatek A

Appendix 1

Spis rysunków

Spis wzorów

Spis algorytmów