Wrocław, 26 maja 2021

Zaawansowane **T**echniki **O**ptymalizacji

Laboratorium 7

Ocena i wizualizacja rozwiązań wielokryterialnych

prowadzący: *dr inż. Jarosław Rudy*

1 Cel laboratorium

Celem laboratorium jest zapoznanie się z specyfiką i problemami rozwiązywania wybranych zadań optymalizacji wielokryterialnej. Zagadnienie obejmuje konstrukcję zbiorów Pareto, skalaryzację kryteriów, obliczanie współczynnika hiperobjętości oraz metody wizualizacji rozwiązań.

2 Przebieg zajęć

Laboratorium obejmuje zajęcia nr 13 i 14 (4 godziny zajęć) i składa się z 3 części (zadań). Praca odbywa się w ramach grup dwuosobowych. Każda grupa otrzymuje do zrealizowania 4 kryteria optymalizacyjne dla wskazanego problemu oraz 4 metody wizualizacji z poniższych list, przy czym:

- w zadaniu pierwszym należy wykorzystać *dwa pierwsze* z podanych 4 kryteriów,
- w zadaniu drugim należy wykorzystać *trzy pierwsze* z podanych 4 kryteriów,
- w zadaniu trzecim należy wykorzystać *wszystkie* podane 4 kryteria oraz wszystkie 4 metody wizualizacji.

Lista kryteriów optymalizacji:

1. Czas zakończenia wszystkich zadań (makespan).
2. Suma czasów zakończenia wszystkich zadań (total flowtime).
3. Maksymalne spóźnienie zadania (max tardiness).
4. Suma spóźnień zadań (total tardiness).
5. Maksymalna nieterminowość zadania (max lateness).
6. Suma nieterminowości zadań (total lateness).

Lista metod wizualizacji:

1. Wykresy słupkowe (bar plots),
2. Ścieżki wartości (value paths),
3. Wykresy kropkowe (multiway dotplots),
4. Współrzędne gwiazdowe (star coordinate system),

5. Odcinkowe współrzędne gwiazdowe (star coordinate system with line segments),

6. Wykresy pajęczynowe (spider web charts),

7. Twarze Chernoffa (Chernoff's faces).

Naliczanie oceny zaczynamy od 2.0. Za poprawne (pełne) wykonanie każdej z 3 części można otrzymać +1.0 do oceny.

3 Problem i kryteria optymalizacji

W tym laboratorium rozpatrywanym problemem jest zagadnienie przepływowe dla 3 maszyn z żądanymi terminami zakończenia zadań. Formalnie dane jest $n \in \mathbb{N}_+$ zadań. Czas wykonywania zadania j na maszynie i (gdzie $i \in \{1, 2, 3\}$) wynosi $p_j^i \in \mathbb{N}_+$. Dodatkowo wartość $d_j \in \mathbb{N}_+$ oznacza pożądany termin zakończenia zadania i. W problemie występują poniższe ograniczenia:

- zadanie należy zakończyć na maszynie i zanim będzie można je rozpocząć na maszynie $i + 1$,
- maszyna może wykonywać co najwyżej jedno zadanie naraz.

Należy określić kolejność wykonywania zadań π, gdzie $\pi(j)$ oznacza zadanie, które będzie wykonane jako j-te w kolejności π. Na podstawie π można określić harmonogram tj. czasy zakończenia C, gdzie $C^i_{\pi(j)}$ oznacza termin zakończenia zadania j-tego w kolejności π na maszynie i. Wyznaczenie harmonogramu C na podstawie kolejności π odbywa się zgodnie z poniższymi równaniami:

$$C^i_{\pi(j)} = \max\{C^{i-1}_{\pi(j)}, C^i_{\pi(j-1)}\} + p^i_{\pi(j)}, \qquad \text{dla } j > 1 \wedge i > 1, \tag{1}$$

$$C^i_{\pi(j)} = C^{i-1}_{\pi(j)} + p^i_{\pi(j)}, \qquad \text{dla } j = 1 \wedge i > 1, \tag{2}$$

$$C^i_{\pi(j)} = C^i_{\pi(j-1)} + p^i_{\pi(j)}, \qquad \text{dla } j > 1 \wedge i = 1, \tag{3}$$

$$C^i_{\pi(j)} = p^i_{\pi(j)}, \qquad \text{dla } j = 1 \wedge i = 1. \tag{4}$$

Dla tego problemu można zdefiniować szereg kryteriów. W naszej sytuacji rozważamy następujące kryteria (wszystkie należy minimalizować):

1. Czas zakończenia wszystkich zadań (makespan)

$$C_{\max} = \max_{j \in \{1,2,\ldots,n\}} C^3_j, \tag{5}$$

2. Suma czasów zakończenia wszystkich zadań (total flowtime)

$$\Sigma F = \sum_{j \in \{1,2,\ldots,n\}} C^3_j, \tag{6}$$

3. Maksymalne spóźnienie zadania (max tardiness)

$$T_{\max} = \max_{j \in \{1,2,\ldots,n\}} \max\{C^3_j - d_j, 0\}, \tag{7}$$

4. Suma spóźnień zadań (total tardiness)

$$\Sigma T = \sum_{j \in \{1,2,\ldots,n\}} \max\{C^3_j - d_j, 0\}, \tag{8}$$

5. Maksymalna nieterminowość zadania (max lateness)

$$L_{\max} = \max_{j \in \{1,2,\ldots,n\}} C_j^3 - d_j, \tag{9}$$

6. Suma nieterminowości zadań (total lateness)

$$\Sigma L = \sum_{j \in \{1,2,\ldots,n\}} C_j^3 - d_j. \tag{10}$$

Generacja instancji Dla parametru n oraz ziarna Z:

1. init(Z).
2. $A = 0$.
3. Dla i od 1 do 3:
   - 3.1. Dla j od 1 do n:
     - 3.1.1. $p_j^i \leftarrow \text{nextInt}(1, 99)$.
     - 3.1.2. $A = A + p_j^i$.
4. $B = \lfloor \frac{1}{2} A \rfloor$.
5. $A = \lfloor \frac{1}{6} A \rfloor$.
6. Dla j od 1 do n:
   - 6.1. $d_j \leftarrow \text{nextInt}(A, B)$.

4 Zadanie 1

W pierwszej części zadania należy napisać prosty algorytm bazujący na uproszczonym schemacie symulowanego wyżarzania (Simulated Annealing) tj. zaczynamy od pewnego rozwiązania zgodnie z poniższym pseudokodem:

1. $P \leftarrow \varnothing$.
2. $i \leftarrow 0$.
3. Ustal początkowe rozwiązanie x (np. losowo).
4. Dodaj x do P.
5. Dopóki $it < maxIter$:
   - 5.1. Wyznacz x' jako losowego sąsiada x.
   - 5.2. Jeśli $x' \prec x$, to wykonaj $x \leftarrow x'$ oraz dodaj x' do P.
   - 5.3. W przeciwnym razie wykonaj $x \leftarrow x'$ oraz dodaj x' do P z prawdopodobieństwem $p(it)$.
   - 5.4. $it \leftarrow it + 1$.
6. Wyznacz front Pareto F ze zbioru P.
7. Wypisz F oraz P.

W powyższym kodzie *maxIter* jest limitem liczby iteracji, zaś $p(it)$ jest prawdopodobieństwem akceptacji. Może być ono stałe (np. $p(it) = 0.1$) lub malejące geometrycznie (np. $p(it) = 0.995^{it}$).

Z kolei $a \prec b$ jest relacją dominacji. Ściślej, zakładając minimalizację k kryteriów, rozwiązanie a dominuje rozwiązanie b (co zapisujemy $a \prec b$) wtedy i tylko wtedy gdy:

$$\forall_{i \in \{1,2,\ldots,k\}} : \; c_i(a) \leqslant c_i(b), \tag{11}$$

$$\exists_{i \in \{1,2,\ldots,k\}} : \; c_i(a) < c_i(b). \tag{12}$$

gdzie $c_i(x)$ to wartość funkcji celu rozwiązania x dla i-tego kryterium. Innymi słowy rozwiązanie a dominuje rozwiązanie b wtedy i tylko wtedy gdy a nie jest na żadnym kryterium gorsze od b i jednocześnie a jest na co najmniej jednym kryterium lepsze od b.

Algorytm wymaga stworzenia m.in. (1) funkcji ruchu (do generacji sąsiadów, najprościej wykorzystać sąsiedztwa insert lub swap), (2) funkcji obliczającej harmonogram (wartości C_j^i) oraz (3) funkcji obliczających wartości obu kryteriów (tzn. wartość funkcji celu jest parą liczb).

Ostatnim elementem algorytmu jest wydzielenie frontu Pareto F ze zboru rozwiązań Pareto P. Polega to na usunięciu ze zbioru P rozwiązań zdominowanych.

1. $F \leftarrow P$.
2. Dla każdego $a \in F$:
   2.1. Dla każdego $b \in F$, takiego że $a \neq b$:
      2.1.1. jeśli $b \prec a$, to wykonaj $F \leftarrow F \setminus \{a\}$ oraz `break`.

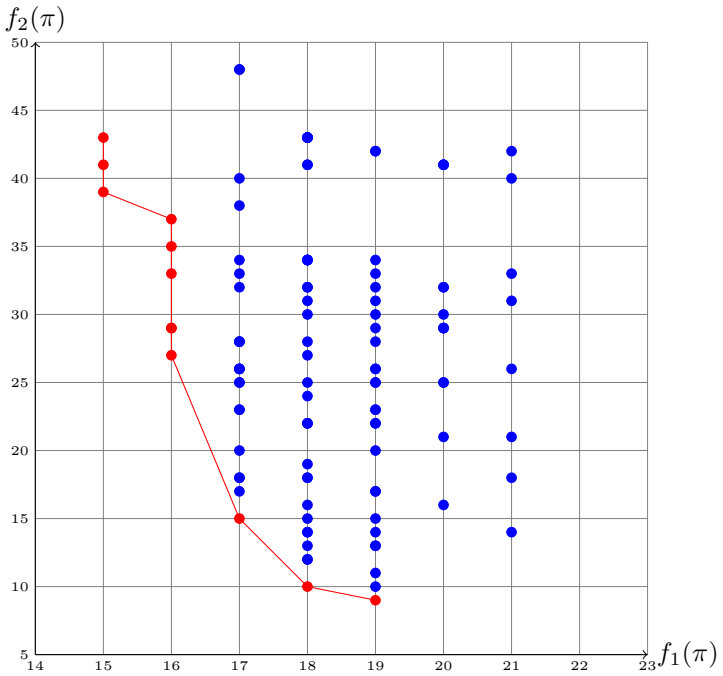
Tak więc w zbiorze F zostają tylko rozwiązania takie że dla dowolnej ich pary a i b nie zachodzi ani $a \prec b$ ani $b \prec a$. Powyższe rozwiązanie jest poprawne, ale można również tworzyć front Pareto na bieżąco, tj. przy dodawaniu x' do P w punktach 5.2 oraz 5.3 można usuwać z P rozwiązania zdominowane przez x' oraz w ogóle nie dodawać x' jeśli któreś z obecnych rozwiązań w P je dominuje. Wtedy P jest na końcu poszukiwanym frontem F.

Używając powyższego algorytmu należy wykonać badania jakości algorytmu (uzyskiwanych frontów Pareto) dla różnych wartości *maxIter* (np. 100, 200, 400, 800, 1600). Oznacza to:

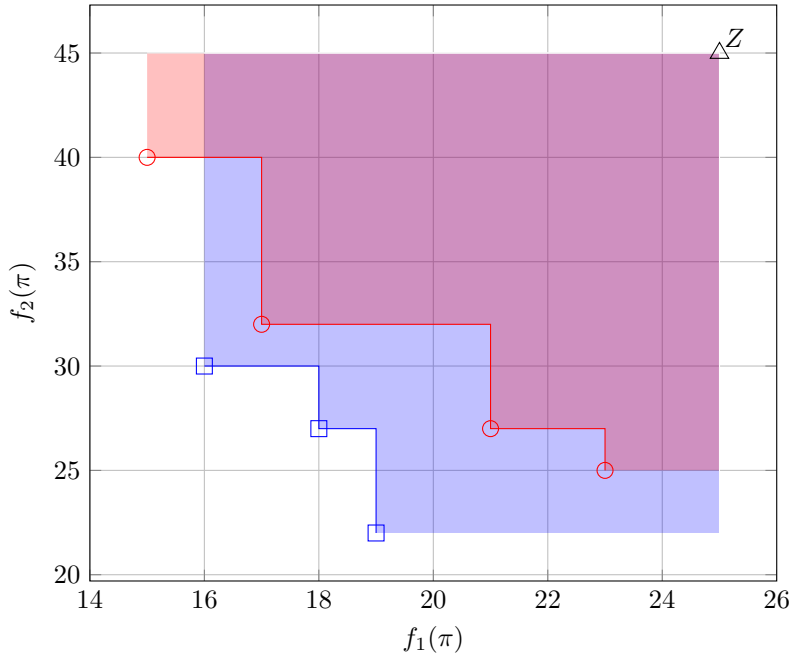
1. Stworzenie dla każdej badanej wartości *maxIter* wykresu zbiorów P oraz F tak że na osi X jest wartość pierwszego kryterium, zaś na osi Y drugiego, wraz z zaznaczeniem frontu Pareto. Aby móc stworzyć taki wykres algorytm musi zwracać oba zbiory (albo samodzielnie wyliczać F na bazie P). Przykład takiego wykresu jest na Rysunku 1.
2. Obliczenie (poadnie wartości) współczynnika hiperobjętości (Hyper Volume Indicator, HVI) dla każdego uzyskanego frontu Pareto (tj. dla każdej badanej wartości *maxIter*).

Aby obliczyć HVI należy w pierwszej kolejności określić punkt referencyjny Z tzw. nadir. Dla 2 kryteriów jest to punkt o współrzędnych (z_1, z_2). Za wartość z_1 przyjmujemy najgorszą (największą) wartość pierwszego kryterium we *wszystkich* rozważanych frontach (dla 5 wartości *maxIter* oznacza to 5 frontów), zwykle pomnożoną przez jakiś współczynnik (zwykle przyjmuje się wartości 1.2 lub 1.0). Podobnie postępujemy dla z_2 na podstawie najgorszej wartości drugiego kryterium we wszystkich porównywanych frontach. Wartość HVI dla danego frontu równa jest polu figury ograniczonej punktami danego frontu i punktem Z, jak przedstawiono to na Rysunku 2.

Uwaga! Ze względu na losowy charakter algorytmu, wyniki HVI najlepiej przedstawić jako średnią z np. 10 przebiegów. Tzn. powyższy proces wykonać 10 razy (za każdym razem uzyskując wartości HVI dla wszystkich badanych *maxIter*), po czym dla każdej wartości *maxIter* podajemy średnią z 10 HVI dla tej liczby iteracji.



Rysunek 1: Przykład wykresu zbioru i frontu Pareto dla dwóch kryteriów



Rysunek 2: Reprezentacja graficzna współczynnika HVI na przykładzie dwóch frontów. Front oznaczony kwadratami posiada większą wartość współczynnika

5 Zadanie 2

W tym zadaniu optymalizacji podlegają 3 kryteria. Należy napisać algorytm zbliżony do tego z zadania poprzedniego, ale z wykorzystaniem skalaryzacji kryteriów. Skalaryzacja polega na zredukowaniu problemu optymalizacji wielokryterialnej do problemu optymalizacji jednokryterialnej: wartością funkcji celu jest pojedyncza liczba s (skalar) wyliczana na podstawie wartości wszystkich k kryteriów x_i do x_k rozwiązania x, najczęściej jako kombinacja liniowa:

$$s(x) = c_1 x_1 + c_2 x_2 + \ldots + c_k x_k. \tag{13}$$

Wartości c_1 do c_n są współczynnikami skalaryzacji, które należy dobrać. Prowadzi to do normalizacji kryteriów. Np. jeśli kryteria mają być równoważne, a wartości x_2 są typowo 5 razy większe od wartości x_1 to współczynniki należy tak dobrać by $c_1 = 5c_2$.

1. $i \leftarrow 0$.
2. Ustal początkowe rozwiązanie x (np. losowo).
3. Wykonaj $x_{\text{best}} \leftarrow s(x)$.
4. Dopóki $it < maxIter$:
   4.1. Wyznacz x' jako losowego sąsiada x.
   4.2. Jeśli $s(x') < s(x)$, to wykonaj $x \leftarrow x'$.
   4.3. W przeciwnym razie wykonaj $x \leftarrow x'$ z prawdopodobieństwem $p(it)$.
   4.4. $it \leftarrow it + 1$.
5. Wypisz x.

Po napisaniu algorytmu należy zbadać jego jakość w zależności od parametru *maxIter* (czyli jak jakość rozwiązania wzrasta wraz z liczbą iteracji). Należy wykorzystać te same instancje i wartości *maxIter* jak dla zadania 1. Dla każdej badanej wartości *maxIter* należy podać wartość $s(x)$. Dodatkowo, można zbadać wpływ różnych wartości parametrów c_1 do c_n na jakość wyników.

Uwaga! Tak jak poprzednio ze względu na losowość algorytmu najlepiej przedstawić wyniki dla 10 przebiegów.

6 Zadanie 3

Zadanie polega na wizualizacji kilku rozwiązań wielokryterialnych używając zadanych metod wizualizacji. Należy rozważyć 4 rozwiązania:

- Trzy rozwiązania pochodzące z końcowego frontu Pareto uzyskanego tym samym algorytmem co w zadaniu nr 1 (tylko że dla czterech zamiast dwóch kryteriów). Jeśli otrzymany front ma więcej niż 3 rozwiązania, to część należy usunąć. Jeśli ma mniej niż 3, to należy wygenerować brakujące. Najlepiej zrobić to poprzez modyfikację (kilka losowych ruchów) któregoś rozwiązań z frontu lub poprzez wykorzystanie któregoś z rozwiązań z wcześniejszych iteracji.
- Jedno rozwiązanie "słabsze". Najprościej jest przyjąć początkowe rozwiązanie algorytmu, przy czym jeśli nie będzie ono zbytnio różniło się jakościowo, to należy rozwiązanie wygenerować osobno (np. losowo).

Powyższe rozwiązania należy przedstawić używając metod wizualizacji zadanych przez prowadzącego. Na koniec należy dla każdej metody wizualizacji dokonać decyzji tj. wyboru rozwiązania. Metody wizualizacji opisane są poniżej.

	kryt. 1	kryt. 2	kryt. 3	kryt. 4
rozwiązanie 1	10	45	30	60
rozwiązanie 2	25	15	35	30
rozwiązanie 3	40	20	15	45
zakres danych	$[10, 60]$	$[10, 50]$	$[10, 40]$	$[20, 60]$

Tabela 1: Przykładowe rozwiązania i zakresy dla problemu optymalizacji dla 4 kryteriów

7 Wizualizacja rozwiązań wielokryterialnych

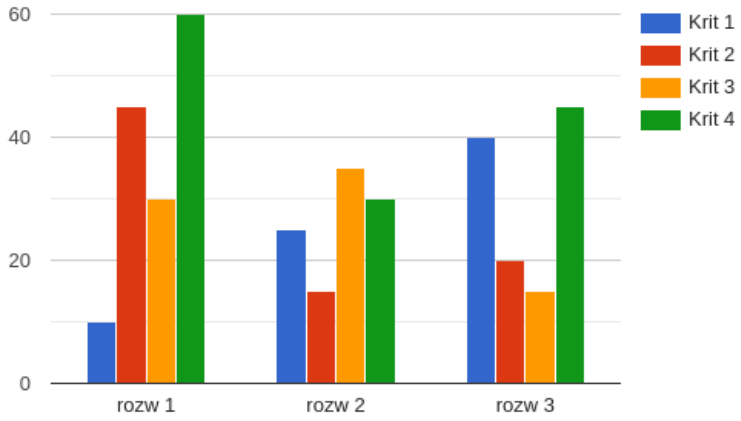
W Tabeli 1 przedstawiono przykładowe trzy rozwiązania dla problemu z czterema kryteriami optymalizacji. Ponadto dla każdego z kryteriów przedstawiony jest zakres wartości. Zakres ten powinien obejmować wszystkie rozwiązania z frontu Pareto (plus w naszym przypadku rozwiązanie "słabsze"). Czyli jeśli front Pareto miał 6 rozwiązań, to zakres ustalamy w oparciu o 7 rozwiązań (6 oraz "słabsze"), pomimo że w zadaniu będziemy porównywać tylko 4 rozwiązania. Dane te posłużą nam jako przykład do wyjaśnienia działania poszczególnych metod wizualizacji.

Uwaga! W przypadku kryteriów, które posiadają zupełnie inne zakresy wartości, może być konieczna normalizacja danych, niezależnie od metody wizualizacji.

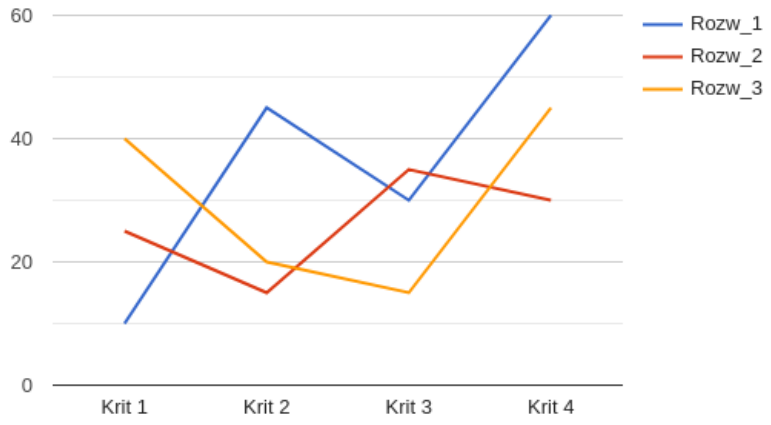
7.1 Wykresy słupkowe

Jest to dosyć prosta metoda wizualizacji. Wszystkie wartości kryteriów (w naszym przypadku jest to 12 wartości) przedstawia się jako pionowe słupki, z wartościami poszczególnych kryteriów na osi Y. Słupki grupuje się albo według kryterium, albo według rozwiązania, zależnie od preferencji. Łatwiejsza w implementacji wydaje się opcja druga. Możliwa jest też opcja z zamianą osi miejscami (wtedy słupki są poziome).

Do utworzenia wykresów można wykorzystać np. stronę https://www.rapidtables.com/tools/



Rysunek 3: Przykład wykresów słupkowych dla danych z Tabeli 1



Rysunek 4: Przykład ścieżek wartości dla danych z Tabeli 1

bar-graph.html lub inne narzędzie do rysowania wykresów (MS Excel, tikz, gnuplot, pakiety `python` itp.).

7.2 Ścieżki wartości

Metoda ta jest podobna do powyższej, ale zamiast wykresu słupkowego, tworzy się wykresy liniowe. Jedną linią łączy się albo kryteriaMetoda ta jest podobna do powyższej, ale zamiast wykresu słupkowego, tworzy się wykresy liniowe. Jedną linią łączy się albo kryteria tego samego rozwiązania (jak na Rysunku 4), albo rozwiązania dla tego samego kryterium. Czasami do tego wykresu dołącza się pionowe słupki które ukazują rozpiętość danego kryterium (całego rozważanego frontu) lub danego rozwiązania.

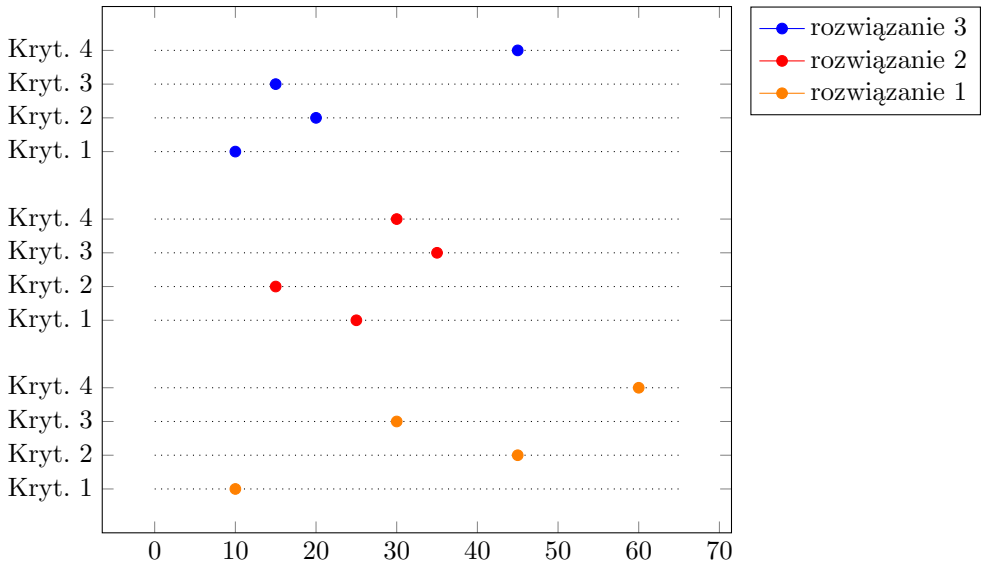
tego samego rozwiązania (jak na Rysunku 4), albo rozwiązania dla tego samego kryterium. Czasami do tego wykresu dołącza się pionowe słupki które ukazują rozpiętość danego kryterium (całego rozważanego frontu) lub danego rozwiązania.

Do utworzenia wykresów można wykorzystać np. stronę https://www.rapidtables.com/tools/bar-graph.html, chociaż nie pozwala ona łączyć wykresów słupkowych z liniowymi. Można wykorzystać też inne narzędzia do rysowania wykresów (MS Excel, tikz, gnuplot, pakiety `python` itp.).

7.3 Wykresy kropkowe

W tego typu wykresie każdemu rozwiązaniu odpowiadają poziome linie (zwykle szare, kropkowane lub kreskowane), tyle ile jest kryteriów, czyli w naszym przypadku linii jest 12, po 4 na rozwiązanie. Każda linia działa podobnie jak oś liczbowa. Na każdej takie osi zaznaczamy kropką jeden punkt przedstawiający wartość danego kryterium dla danego rozwiązania. Linie stanowiące jedno rozwiązanie umieszczone są jedna po drugiej, zaś większa przerwa umieszczana jest pomiędzy rozwiązaniami. Przykładowy wygląd został przedstawiony na Rysunku 5.

Do utworzenia wykresów można wykorzystać dowolne narzędzie, które pozwala łączyć wykresy punktowe z liniowymi (oraz wybrać styl linii). Zalecanymi rozwiązaniami jest tikz lub gnuplot.



Rysunek 5: Przykład wykresu kropkowego dla danych z Tabeli 1

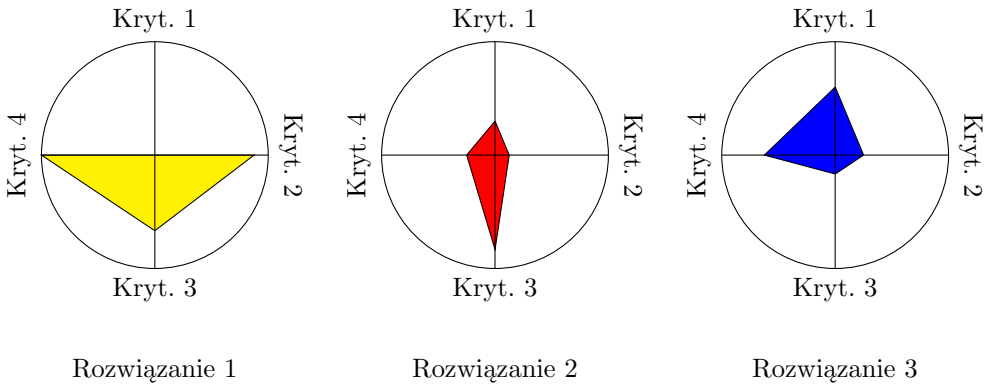
7.4 Współrzędne gwiazdowe

W tej metodzie dla k kryteriów rysuje się okręgi, po jednym dla każdego rozwiązania. Wewnątrz okręgu rysuje się k promieni równomiernie rozłożonych (tak by punkty zetknięcia promieni z okręgiem stanowiły wierzchołki wielokąta foremnego o k wierzchołkach). Następnie na i-tym promieniu umieszcza się punkt odpowiadający wartości i-tego kryterium znormalizowanej do przedziału $[0; 1]$. Innymi słowy, jeśli wartość jest najmniejsza w całym rozważanym froncie (dla tego kryterium), to punkt rysujemy w centrum okręgu, jeśli najgorszy to na obwodzie okręgu, zaś wartości pośrednie wzdłuż promienia, im wartość gorsza, tym dalej od środka okręgu. Utworzone w ten sposób punkty tworzą wierzchołki wielokąta (zwykle mającego k wierzchołków, ale może być ich mniej jeśli pojawią się punkty współliniowe).

W tej sytuacji rozwiązanie "idealne" byłoby punktem, zaś rozwiązanie "najgorsze" miałoby wszystkie wierzchołki na obwodzie okręgu. Oznacza to że lepsze rozwiązania zwykle odpowiadają mniejszemu polu powierzchni wielokąta ("gwiazdy"). Oczywiście można sytuację odwrócić (1 oznacza środek, 0 obwód), by większe pole sugerowało lepsze rozwiązanie. Zwykle uzyskane wielokąty wypełnia się kolorem, choć spotyka się też wersje niewypełnione. Możliwe jest też nałożenie kilku rozwiązań w ramach jednego okręgu. Na koniec należy zaznaczyć, że metoda ta (kształt i pole powierzchni wielokątów) jest wrażliwa na kolejność kryteriów! Możliwa realizacja dla przykładowych danych została przedstawiona na Rysunku 6. Do stworzenia wykresu współrzędnych gwiazdowych można wykorzystać narzędzia LaTeX i `tikz` udostępniane przez prowadzącego.

7.5 Odcinkowe współrzędne gwiazdowe

Jest to metoda bardzo zbliżona do poprzedniej. Jednakże po uzyskaniu punktów wzdłuż każdego promienia, zamiast łączyć je w wielokąt, rysujemy odcinki od nich do środka okręgu. W tym przypadku długość odcinków informuje o jakości danego kryterium. Podobnie jak w poprzedniej metodzie można odwrócić znaczenie (wtedy dłuższe odcinki oznaczają lepszą wartość). Metoda ta jest mniej wrażliwa na kolejność kryteriów. Możliwa realizacja dla przykładowych danych została przedstawiona na Rysunku 6. Podobnie jak poprzednie do wizualizacji można wykorzystać narzędzia LaTeX i `tikz` udostępniane przez prowadzącego.

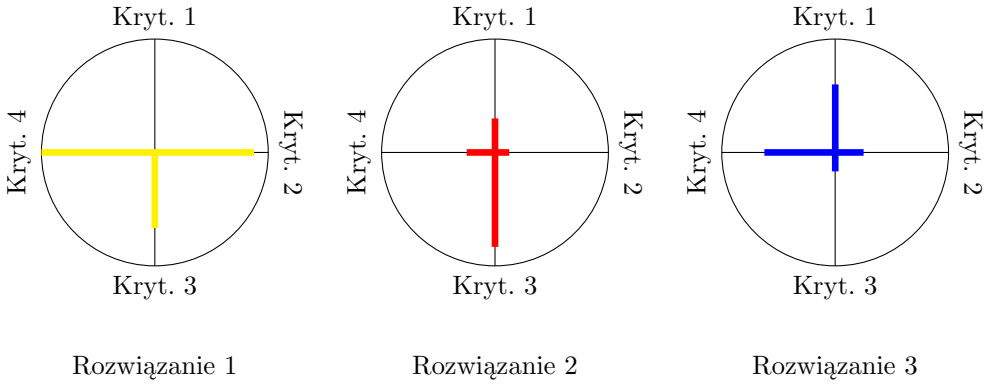


Rysunek 6: Przykład wizualizacji współrzędnych gwiazdowych dla danych z Tabeli 1

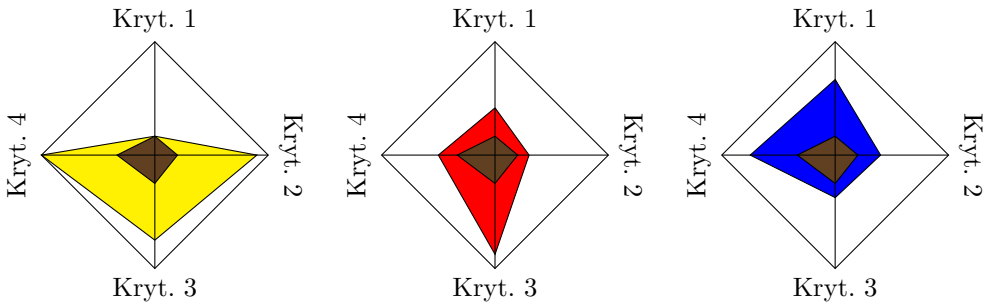
7.6 Wykresy pajęczynowe

Metoda ta jest zbliżona do współrzędnych gwiazdowych, z kilkoma różnicami. Pierwsza różnica dotyczy normalizacji. We współrzędnych gwiazdowych środek okręgu oznaczał najlepszą wartość danego kryterium w całym froncie. W wykresie pajęczynowym natomiast środek okręgu oznacza wartość 0. Oznacza to, że środek okręgu może być nieosiągalny jeśli funkcja celu nie przyjmuje wartości 0. Drugą różnicą jest to że "obwód" nie jest przedstawiany w postaci okręgu, ale wielokąta foremnego (dla zachowania podobieństwa do sieci pajęczej). Ostatnia różnica to dodatkowy wielokąt umieszczany w centrum sieci, obrazujący "obszar nieosiągalny".

Jedną z wad tej wizualizacji jest sytuacja, gdy wartości któregoś kryterium są znaczne, ale o niewielkim rozrzucie. Np. jeśli wszystkie kryteria będą miały zakres $[1100; 1200]$ to "obszar nieosiągalny" wypełni przeszło 90% wykresu. Wadę tę można zniwelować, zmieniając znaczenie środka wykresu, tak by nie oznaczał on zera, ale wartości rozwiązania idealnego. Przykładowo dla kryterium ΣF (total flowtime) można obliczyć sumę wszystkich operacji. Taka suma stanowi dolne ograniczenie wartości ΣF i obrazuje niejako przypadek idealny, nawet jeśli wszystkie wartości tego kryterium w rozważanym froncie są znacznie większe. Możliwa realizacja dla przykładowych danych została przedstawiona na Rysunku 8. Podobnie jak poprzednie do wizualizacji można wykorzystać narzędzia LaTeX i `tikz` udostępniane przez prowadzącego.



Rysunek 7: Przykład wizualizacji odcinkowych współrzędnych gwiazdowych dla danych z Tabeli 1



Rozwiązanie 1 Rozwiązanie 2 Rozwiązanie 3

Rysunek 8: Przykład wizualizacji wykresów pajęczynowych dla danych z Tabeli 1

7.7 Twarze Chernoffa

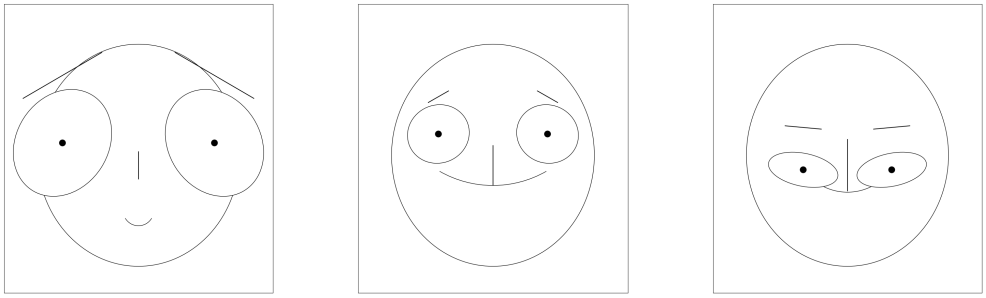
W tej metodzie (która może być stosowana nawet dla 18 kryteriów) wartości poszczególnych kryteriów wpływają na cechy twarzy takie jak:

- kształt twarzy,
- rozmiar, kształt i umiejscowienie oczu,
- rozmiar i wykrzywienie ust,
- rozmiar nosa,
- rozmiar i nachylenie brwi.

Wybór rozwiązania jest kwestią subiektywnej oceny estetyki lub nastroju twarzy – decydent wybiera rozwiązanie, które odpowiada najbardziej podobającej mu się twarzy.

W celu wykonania ćwiczenia dostępny jest skrypt generujący twarze Chernoffa w języku `python`. Użycie polega na wywołaniu funkcji `drawFace`, jako parametry podając znormalizowane wartości 4 kryteriów. Należy przedtem rozwiązać jednak dwie kwestie:

- które kryterium odpowiada któremu parametrowi (24 możliwości ułożenia dla 4 kryteriów),
- normalizacja kryteriów, tak by każde miało wartości w przedziale od 0 do 1.



(a) Rozwiązanie 1 (b) Rozwiązanie 2 (c) Rozwiązanie 3

Rysunek 9: Przykład twarzy Chernoffa dla danych z Tabeli 1

Możliwa realizacja twarzy Chernoffa dla przykładowych danych widoczna jest na Rysunku 9.