Programowanie celowe #1

- Problem programowania celowego (PC) jest przykładem problemu programowania matematycznego nieliniowego, który można skutecznie zlinearyzować, tzn. zapisać (i rozwiązać) jako problem programowania liniowego

Programowanie celowe #2

- Ogólna (nieważona) postać problemów PC:

$$\min \sum_{i=1}^{S} |\mathbf{c}_i^T \mathbf{x} - c_i|$$

p.o.: **Ax** {≤, ≥, **=**} **b**

x ≥ **0**

– cechy problemu
  - kierunek optymalizacji w funkcji celu: zawsze minimalizacja
  - skalarne współczynniki c_i występujące w funkcji celu nazywane są celami, wymaga się aby wartości wyrażeń $\mathbf{c}_i^T \mathbf{x}$ dążyły do odpowiednich celów (czyli wartości c_i)
  - najmniejszą możliwą wartością funkcji celu jest wartość 0
    – wartość ta jest uzyskiwana wtedy i tylko wtedy, gdy dla każdego i zachodzi: $\mathbf{c}_i^T \mathbf{x} = c_i$

Programowanie celowe #3

- Ogólna, ważona postać problemów PC:

$$\min \sum_{i=1}^{S} w_i \,|\, \mathbf{c}_i^T \mathbf{x} - \mathrm{c}_i \,|$$

p.o.: **Ax** {≤, ≥, =} **b**

x ≥ **0**

– cechy problemu (oprócz wymienionych poprzednio)
  - wartość funkcji celu jest kontrolowana za pomocą wag w_i, których działanie jest następujące:
    – im większa wartość wagi w_i tym większa waga przykładana jest do osiągnięcia celu c_i (kosztem dokładności osiągania innych celów)

Programowanie celowe #4

- Wyprowadzenie PC – rozwiązywanie układów równań
  - rozważmy oznaczony układ równań postaci **Ax**=**b**

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$
  - zgodnie z twierdzeniem Kroneckera–Capellie'go
    - rozwiązanie tego problemu istnieje ponieważ
      - rank(**A**)=rank([**A b**]

        (jest to układ dwóch równań z dwoma niewiadomymi)
    - rozwiązanie to jest jednoznaczne ponieważ
      - rank(**A**)=N, gdzie N jest liczbą kolumn macierzy **A**

        (równania te są niezależne)
    - rozwiązanie: wektor **x**=$[1\ 2]^T$

Programowanie celowe #5

- Wyprowadzenie PC – rozwiązywanie układów równań
  - rozważmy nieoznaczony układ równań postaci **A**x=**b**

    (1) $1x_1+2x_2=5$

    (2) $2x_1+4x_2=10$
  - zgodnie z twierdzeniem Kroneckera–Capellie'go
    - rozwiązanie tego problemu istnieje ponieważ
      - rank(**A**)=rank([**A b**]

        (jest to układ dwóch równań z dwoma niewiadomymi)
    - rozwiązanie to jest niejednoznaczne ponieważ
      - rank(**A**)<N, gdzie N jest liczbą kolumn macierzy **A**

        (równania te są zależne)
    - rozwiązanie: każdy wektor postaci **x**=$[5–2p, p]^T$, gdzie $p\in(-\infty,+\infty)$

Programowanie celowe #6

- Wyprowadzenie PC – rozwiązywanie układów równań
  - rozważmy sprzeczny układ równań postaci **Ax**=**b**

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$

    (3) $2x_1+1x_2=5$
  - zgodnie z twierdzeniem Kroneckera–Capellie'go
    - rozwiązanie tego problemu nie istnieje ponieważ
      - rank(**A**)<rank([**A b**]

        (jest to układ trzech niezależnych równań z dwoma niewiadomymi)
  - powstaje pytanie, czy jeżeli dokładne rozwiązanie tego problemu nie istnieje, to czy można zaproponować jakieś rozwiązanie przybliżone?

Programowanie celowe #7

- Wyprowadzenie PC – rozwiązanie przybliżone
  - pomysły na uzyskanie rozwiązania przybliżonego: tworzenie różnych par równań i ich rozwiązanie, a konkretnie:
    - równanie pierwsze i drugie tworzą układ

(1) $1x_1+2x_2=5$

(2) $3x_1+2x_2=7$

– którego rozwiązaniem jest: $\mathbf{x}=[1, 2]^T$

- równanie pierwsze i trzecie tworzą układ:

(1) $1x_1+2x_2=5$

(3) $2x_1+1x_2=5$

– którego rozwiązaniem jest: $\mathbf{x}=[5/3, 5/3]^T$

- równanie drugie i trzecie tworzą układ:

(2) $3x_1+2x_2=7$

(3) $2x_1+1x_2=5$

– którego rozwiązaniem jest: $\mathbf{x}=[3, -1]^T$

Programowanie celowe #8

- Wyprowadzenie PC – rozwiązanie przybliżone
  - można powiedzieć więc, że układ równań posiada trzy rozwiązania przybliżone:
    - rozwiązanie: $\mathbf{x}=[1, 2]^T$
    - rozwiązanie: $\mathbf{x}=[5/3, 5/3]^T$
    - rozwiązanie: $\mathbf{x}=[3, -1]^T$
  - powstają pytania, czy:
    - rozwiązania te są równie dobre, czy też może różnią się od siebie?
    - a jeżeli się różnią, to czym i które z nich jest najlepsze?

Programowanie celowe #9

- Wyprowadzenie PC – rozwiązanie przybliżone
  - otrzymane rozwiązania można scharakteryzować następująco:
    - **x**=$[1, 2]^T$ spełnia równania (1) i (2) a nie spełnia równania (3)
    - **x**=$[5/3, 5/3]^T$ spełnia równania (1) i (3) a nie spełnia równania (2)
    - **x**=$[3, -1]^T$ spełnia równania (2) i (3) a nie spełnia równania (1)
  - jeżeli jakieś rozwiązanie **x** nie spełnia równania $\mathbf{c}^T\mathbf{x}=c$, to znaczy, że wartością wyrażenia $\mathbf{c}^T\mathbf{x}$ jest d, przy czym $d \neq c$
  - rozwiązanie można więc ocenić obliczając różnicę pomiędzy wartościami d oraz c (a konkretnie: wartość bezwzględną z tej różnicy)
    - wartość tę nazwiemy odchyłką
    - oczywiście im mniejsza odchyłka tym lepsze rozwiązanie

Programowanie celowe #10

- Wyprowadzenie PC – rozwiązanie przybliżone
  - otrzymane rozwiązania można scharakteryzować następująco:
    - rozwiązanie $\mathbf{x}=[1, 2]^T$
      - nie spełnia równania (3), czyli równania $2x_1+1x_2=5$, mamy więc: $2\cdot1+1\cdot2=4$ (choć powinno być 5), odchyłka: $|4–5|=1$
    - rozwiązanie $\mathbf{x}=[5/3, 5/3]^T$
      - nie spełnia równania (2), czyli równania $3x_1+2x_2=7$, mamy więc: $3\cdot5/3+2\cdot5/3=25/3$ (choć powinno być 7 //czyli 21/3//), odchyłka: $|25/3–21/3|=4/3$
    - rozwiązanie $\mathbf{x}=[3, –1]^T$
      - nie spełnia równania (1), czyli równania $1x_1+2x_2=5$, mamy więc: $1\cdot3+2\cdot(–1) = 1$ (choć powinno być 5), odchyłka: $|1–5|=4$
  - widać więc, że za najlepsze rozwiązanie można uznać rozwiązanie $\mathbf{x}=[1, 2]^T$ (ponieważ jego odchyłka jest najmniejsza)

Programowanie celowe #11

- Wyprowadzenie PC – rozwiązanie przybliżone
  - odchyłki mogą być jednak obliczone ogólniej (dzięki czemu będzie można oceniać dowolne inne rozwiązania, a nie tylko takie, które powstały w rezultacie rozwiązywania wybranych par równań
  - odchyłkę definiujemy jako sumę wartości bezwzględnych z różnic pomiędzy lewymi a prawymi stronami równań (oczywiście po wstawieniu do lewych stron nich ocenianych rozwiązań)

Programowanie celowe #12

- Wyprowadzenie PC – rozwiązanie przybliżone
  - sumaryczne odchyłki rozważanych rozwiązań
    - rozwiązanie **x**=[1, 2]^T

$$|1\cdot1+2\cdot2-5|+|3\cdot1+2\cdot2-7|+|2\cdot1+1\cdot2-5| = |5-5|+|7-7|+|4-5| =$$
$$= 0+0+1 = 1$$

    - rozwiązanie **x**=[5/3, 5/3]^T

$$|1\cdot5/3+2\cdot5/3-5|+|3\cdot5/3+2\cdot5/3-7|+|2\cdot5/3+1\cdot5/3-5| =$$
$$= |5-5|+|25/3-7|+|5-5| = 0+4/3+0 = 4/3$$

    - rozwiązanie **x**=[3, –1]^T

$$|1\cdot3+2\cdot(-1)-5|+|3\cdot3+2\cdot(-1)-7|+|2\cdot3+1\cdot(-1)-5| =$$
$$= |1-5|+|7-7|+|5-5| = 4+0+0 = 4$$

  - oczywiście otrzymane odchyłki są identyczne z poprzednimi, co wynika z faktu, że rozważane rozwiązania spełniają zawsze dwa spośród trzech równań, w rezultacie czego część odchyłki, która odpowiada tym równaniom, będzie zawsze wynosiła 0

Programowanie celowe #13

- Wyprowadzenie PC – rozwiązanie przybliżone
  - dzięki ogólnym odchyłkom można porównać dotychczasowe rozwiązania:
    - rozwiązanie $\mathbf{x}=[1, 2]^T$, odchyłka 1
    - rozwiązanie $\mathbf{x}=[5/3, 5/3]^T$, odchyłka 4/3
    - rozwiązanie $\mathbf{x}=[3, -1]^T$, odchyłka 4
  - z rozwiązaniem znalezionym dowolną inną metodą, np. z rozwiązaniem $\mathbf{x}=[1, 1]^T$, którego odchyłka wynosi:

    $|1 \cdot 1+2 \cdot 1-5|+|3 \cdot 1+2 \cdot 1-7|+|2 \cdot 1+1 \cdot 1-5| = |3-5|+|5-7|+|3-5| =$

    $= 2+2+2 = 6$
    - rozwiązanie to jest więc gorsze od rozwiązań uzyskanych poprzednio

Programowanie celowe #14

- Wyprowadzenie PC – rozwiązanie przybliżone
  - definicja odchyłki sumarycznej nasuwa sposób jeszcze lepszego rozwiązywania postawionego problemu
    - celem jest więc znalezienie rozwiązania, które charakteryzuje się minimalną odchyłką (choć być może nie spełnia w sposób dokładny żadnego z równań)
  - tak postawiony problem może być rozwiązany metodami programowania matematycznego

Programowanie celowe #15

- Wyprowadzenie PC – rozwiązanie przybliżone
  - układ S równań

    (1) $(\mathbf{c}_1)^T\mathbf{x}=c_1$

    ...

    (S) $(\mathbf{c}_S)^T\mathbf{x}=c_S$
  - definicje odchyłek:

    równanie (1): $(\mathbf{c}_1)^T\mathbf{x}=c_1$, odchyłka: $|(\mathbf{c}_1)^T\mathbf{x}-c_1|$

    ...

    równanie (S): $(\mathbf{c}_S)^T\mathbf{x}=c_S$, odchyłka: $|(\mathbf{c}_1)^T\mathbf{x}-c_S|$
  - suma odchyłek:

    $|(\mathbf{c}_1)^T\mathbf{x}-c_1|+|(\mathbf{c}_2)^T\mathbf{x}-c_2|+...+|(\mathbf{c}_S)^T\mathbf{x}-c_S| = \sum_{i=1}^{S}|\mathbf{c}_i^T\mathbf{x}-c_i|$

Programowanie celowe #16

- Wyprowadzenie PC – rozwiązanie przybliżone
  - formalne sformułowanie problemu poszukiwania przybliżonych rozwiązań układów równań więc jest następujące

$$\min \quad \sum_{i=1}^{S} |\, \mathbf{c}_i^T \mathbf{x} - \mathrm{c}_i \,|$$

  - jak więc widać jest to problem bardzo podobny do problemu PC (różnica: brak ograniczeń na nieujemność zmiennych)
  - problem PC można więc traktować jako metodę służącą do przybliżonego rozwiązywania układów równań liniowych

Programowanie celowe #17

- Wyprowadzenie PC – rozwiązanie przybliżone
  - po dodaniu ograniczeń na nieujemność zmiennych: $\mathbf{x} \geq \mathbf{0}$ oraz dodatkowych (liniowych) ograniczeń roboczych postaci $\mathbf{Ax}$ {≤, ≥, =} $\mathbf{b}$ otrzymujemy pełną postać problemu PC

$$\min \sum_{i=1}^{S} |\mathbf{c}_i^T \mathbf{x} - \mathrm{c}_i|$$

p.o.: $\mathbf{Ax}$ {≤, ≥, =} $\mathbf{b}$

$\mathbf{x} \geq \mathbf{0}$

  - problem programowania celowego można więc traktować jako metodę znajdującą rozwiązania wspólnych układów równań i nierówności liniowych, przy dla części równań poszukuje się w ogólności rozwiązania przybliżonego

Programowanie celowe #18

- Przykłady zadań PC
  - znaleźć rozwiązanie oznaczonego układu równań:

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$

    - rozwiązanie układu $[1, 2]^T$
  - zapis zadania w postaci PC:

    min $|1x_1+2x_2–5|+|3x_1+2x_2–7|$

    p.o. $x\geq0$, $x_2\geq0$

Programowanie celowe #19

- Przykłady zadań PC
  - znaleźć rozwiązanie oznaczonego układu równań:

    (1) $1x_1 – 2x_2 = 5$

    (2) $3x_1 – 2x_2 = 7$

    - rozwiązanie układu $[\ 1, –2\ ]^T$
  - zapis zadania w postaci PC:

    min $|1x_1 – 2x_2 – 5| + |3x_1 – 2x_2 – 7|$

    p.o. $x \geq 0$, $x_2 \geq 0$

Programowanie celowe #20

- Przykłady zadań PC
  - znaleźć nieujemne przybliżenie rozwiązania sprzecznego układu równań:

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$

    (3) $2x_1+1x_2=3$

    - brak rozwiązań
  - zapis zadania w postaci PC:

    min $|1x_1+2x_2-5|+|3x_1+2x_2-7|+|2x_1+1x_2-3|$

    p.o. $x\geq 0$, $x_2\geq 0$

Programowanie celowe #21

- Przykłady zadań PC
  - znaleźć rozwiązanie zadania PC:

    min $|1x_1+2x_2-5|+|3x_1+2x_2-7|+|2x_1+1x_2-3|$

    p.o. $5x_1+7x_2 \leq 10$

    $x \geq 0,\ x_2 \geq 0$

Programowanie celowe #22

- Linearyzacja problemów PC
  – problemy PC należą do grupy problemów programowania matematycznego nieliniowego, który można skutecznie zlinearyzować
    - zlinearyzować czyli zapisać (i rozwiązać) jako problem programowania liniowego, którego rozwiązanie będzie jednocześnie rozwiązaniem problemu nieliniowego (lub rozwiązaniem, na podstawie którego można jednoznacznie odczytać rozwiązanie problemu liniowego)
  – linearyzacja problemu PC została podana przez Charnes'a i Cooper'a
  – linearyzacja ta polega na rozkładzie dowolnych wartości rzeczywistych na różnicę dwóch wartości nieujemnych

Programowanie celowe #23

- Linearyzacja problemów PC – rozkład dowolnej wartości na różnicę dwóch wartości nieujemnych
  - dana jest wartość skalarna $x \in (-\propto, +\propto)$
  - wartość tę można przedstawić w postaci różnicy:

    $x = y - z$
  - której składniki, tzn. y i z są wartościami nieujemnymi, tzn. $y \in [0, +\propto)$ oraz $z \in [0, +\propto)$
  - przykłady:

    $5 = 9 - 4$

    $-12 = 8 - 20$

    $50 = 100 - 50$

    $-1 = 1 - 2$

Programowanie celowe #24

- Linearyzacja problemów PC – rozkład dowolnej wartości na różnicę dwóch wartości nieujemnych
  - z definicji rozkład taki jest niejednoznaczny,
    tzn. dla każdego x istnieje nieskończenie wiele par y i z takich, że x = y – z
  - przykłady:

    5 = 7 – 2

    5 = 8 – 3

    5 = 9 – 4

    5 = 9.5 – 4.5

    5 = 9.789 – 4.789

    itd.

Programowanie celowe #25

- Linearyzacja problemów PC – rozkład dowolnej wartości na różnicę dwóch wartości nieujemnych
  - rozkład ten można „ujednoznacznić” przyjmując, że co najmniej jedna z wartości y oraz z powinna być ustalona i wynosić 0
  - przykłady:
    - przykład liczby dodatniej:

    5 = 5 – 0 // jest to jedyna możliwość rozkładu liczby 5

    - przykład liczby zerowej:

    0 = 0 – 0 // jest to jedyna możliwość rozkładu liczby 0

    - przykład liczby ujemnej:

    –5 = 0 – 5 // jest to jedyna możliwość rozkładu liczby –5

Programowanie celowe #26

- Linearyzacja problemów PC – rozkład dowolnej wartości na różnicę dwóch wartości nieujemnych
  - fakt, że co najmniej jedna z wartości y oraz z musi wynosić 0 można w ogólności zapisać za pomocą warunku:

    $y \cdot z = 0$

    (warunek ten jest prawdziwy wtedy i tylko wtedy gdy co najmniej jedna z wartości y oraz jest jest zerem)
  - ogólny (jednoznaczny) rozkład wartości x można więc „zadać” następująco: dla danej $x \in (-\propto, +\propto)$ znaleźć takie $y \in [0, +\propto)$ oraz $z \in [0, +\propto)$, aby zachodziły warunki:

    $x = y - z$

    oraz

    $y \cdot z = 0$

Programowanie celowe #27

- Linearyzacja problemów PC – rozkład dowolnej wartości na różnicę dwóch wartości nieujemnych
  - dysponowanie wartościami y i z (z których co najmniej jedna jest równa zero) jest korzystne, ponieważ:
    - y i z mogą przyjąć rolę zmiennych w problemie PL (zmienne te są zawsze nieujemne)
    - na podstawie y i z łatwo można obliczyć wartość bezwzględną z wartości x:

      jeżeli $x = y - z$ oraz $y \cdot z = 0$ to $|x| = y + z$
  - przykłady:

    $5 = 5 - 0,\ 5 \cdot 0 = 0,$ czyli $|5| = 5 + 0 = 5$

    $0 = 0 - 0,\ 0 \cdot 0 = 0,$ czyli $|0| = 0 + 0 = 0$

    $-5 = 0 - 5,\ 0 \cdot 5 = 0,$ czyli $|-5| = 0 + 5 = 5$

Programowanie celowe #28

- Linearyzacja problemów PC – zastosowanie rozkładu
  - operację rozkładu dowolnej wartości na składniki nieujemne można zastosować do dowolnego wyrażenia, a nie tylko do pojedynczej wartości, np.:

    jeżeli $3x_1+4x_2–10 = y – z$ oraz $y·z = 0$ to $|3x_1+4x_2–10| = y + z$
  - technikę tę można zastosować w problemie PC, którego funkcja celu jest postaci:
  - min $\sum_{i=1}^{S} |\mathbf{c}_i^T \mathbf{x} - \mathrm{c}_i|$
  - w tym przypadku dla każdego i można znaleźć takie y_i oraz z_i aby zachodziło:

    $(\mathbf{c}_i)^T\mathbf{x}–c_i = y_i – z_i$ oraz $y_i·z_i=0$, wtedy oczywiście $|(\mathbf{c}_i)^T\mathbf{x}–c_i| = y_i + z_i$
    - pozwala to na usunięcie wartości bezwzględnej z funkcji celu

Programowanie celowe #29

- Linearyzacja problemów PC – zastosowanie rozkładu
  - powstaje nowa funkcja celu:
  - min $\sum_{i=1}^{S} (y_i + z_i)$
  - oczywiście należy dodać ograniczenia, które wiążą (występujące w nowej funkcji celu) zmienne y_i i z_i z wyrażeniami $(\mathbf{c}_i)^T\mathbf{x}-c_i$:

    dla każdego i: $(\mathbf{c}_i)^T\mathbf{x}-c_i = y_i - z_i$
  - dodajemy także ograniczenia „ujednoznaczniające" rozkład:

    dla każdego i: $y_i \cdot z_i = 0$

Programowanie celowe #30

- Linearyzacja problemów PC – zastosowanie rozkładu
  - dla kompletności dodajemy oryginalne ograniczenia problemu:

    $\mathbf{Ax}\ \{\leq, \geq, =\}\ \mathbf{b}$
  - w tym ograniczenia na nieujemność zmiennych **x**

    $\mathbf{x} \geq \mathbf{0}$
  - ograniczenia te „rozciągają” się na wszystkie zmienne, łącznie z nowo wprowadzonymi zmiennymi y_i i z_i, co w postaci wektorowej można zapisać następująco:

    $\mathbf{y} \geq \mathbf{0}$

    $\mathbf{z} \geq \mathbf{0}$

Programowanie celowe #31

- Linearyzacja problemów PC – zastosowanie rozkładu
  - w rezultacie otrzymujemy problem:
  - min $\sum_{i=1}^{S}(y_i + z_i)$
  - przy ograniczeniach:

    dla każdego i: $(\mathbf{c}_i)^T\mathbf{x} - c_i = y_i - z_i$

    dla każdego i: $y_i \cdot z_i = 0$

    $\mathbf{Ax}$ {≤, ≥, =} $\mathbf{b}$

    $\mathbf{x} \geq \mathbf{0}$

    $\mathbf{y} \geq \mathbf{0}$

    $\mathbf{z} \geq \mathbf{0}$
  - powstały problem byłby problemem PL gdyby nie nieliniowe ograniczenia postaci: $y_i \cdot z_i = 0$

Programowanie celowe #32

- Linearyzacja problemów PC – zastosowanie rozkładu
  - okazuje się jednak, że ograniczenia $y_i \cdot z_i = 0$ można pominąć (a pomimo tego rozwiązanie optymalne będzie identyczne)
    - jest to cecha charakterystyczna dla tzw. problemów programowania wypukłego, którą można uzasadnić na następującym przykładzie:
    - załóżmy, że wartość pewnego wyrażenia postaci $(\mathbf{c}_i)^T\mathbf{x} - c_i$ wynosi 5
      - wartość 5 można np. przedstawić w postaci różnicy: 5 = 7 – 2 (czyli y=7, z=2), wtedy wyrażenie (y + z) w funkcji celu przyjmie wartość 9
      - gdyby wartość 5 przedstawić w postaci różnicy: 5 = 6 – 1 (czyli y=6, z=1), to wtedy wyrażenie (y + z) w funkcji celu przyjmie wartość 7
      - gdyby wreszcie wartość 5 przedstawić w postaci różnicy: 5 = 5 – 0 (czyli y=5, z=0), to wyrażenie (y + z) w funkcji celu przyjmie wartość 5, co stanowi wartość minimalną tego wyrażenia
    - ponieważ wyrażenie reprezentujące wartość funkcji celu jest minimalizowane, to wartości par zmiennych y i z będą zawsze tak dobierane, aby jedna z nich przyjmowała wartość zero

Programowanie celowe #33

- Linearyzacja problemów PC – zastosowanie rozkładu
  - ostateczna postać powstałego problemu:
  - min $\sum_{i=1}^{S} (y_i + z_i)$
  - przy ograniczeniach:

    dla każdego i: $(\mathbf{c}_i)^T \mathbf{x} - c_i = y_i - z_i$

    $\mathbf{Ax} \{\leq, \geq, =\} \mathbf{b}$

    $\mathbf{x} \geq \mathbf{0}$

    $\mathbf{y} \geq \mathbf{0}$

    $\mathbf{z} \geq \mathbf{0}$
  - powyższy problem jest zlinearyzowaną wersją problemu celowego, czyli takim problemem liniowym, którego rozwiązanie jest identyczne z rozwiązaniem problemu celowego

Programowanie celowe #34

- Rozwiązania przykładowych PC
  - znaleźć rozwiązanie oznaczonego układu równań:

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$

    - rozwiązanie układu $[\ 1, 2\ ]^T$
  - zapis zadania w postaci PC:

    min $|1x_1+2x_2–5|+|3x_1+2x_2–7|$

    p.o. $x\geq0$, $x_2\geq0$

    - rozwiązanie optymalne: $[\ 1, 2\ ]^T$
    - wartość funkcji celu: 0
  - otrzymano identyczne rozwiązania

Programowanie celowe #35

- Rozwiązania przykładowych PC
  - znaleźć rozwiązanie oznaczonego układu równań:

    (1) $1x_1 – 2x_2 = 5$

    (2) $3x_1 – 2x_2 = 7$

    - rozwiązanie układu $[\ 1, –2\ ]^T$
  - zapis zadania w postaci PC:

    min $|1x_1 – 2x_2 – 5| + |3x_1 – 2x_2 – 7|$

    p.o. $x \geq 0$, $x_2 \geq 0$

    - rozwiązanie optymalne: $[\ 2.333..., 0\ ]^T$
    - wartość funkcji celu: 2.666...
  - otrzymane rozwiązania różnią się (ponieważ rozwiązanie otrzymane metodą PC musi być nieujemne)

Programowanie celowe #36

- Rozwiązania przykładowych PC
  - znaleźć nieujemne przybliżenie rozwiązania sprzecznego układu równań:

    (1) $1x_1+2x_2=5$

    (2) $3x_1+2x_2=7$

    (3) $2x_1+1x_2=3$

    - brak rozwiązań
  - zapis zadania w postaci PC:

    min $|1x_1+2x_2–5|+|3x_1+2x_2–7|+|2x_1+1x_2–3|$

    p.o. $x \geq 0$, $x_2 \geq 0$

    - rozwiązanie optymalne: $[\ 1,\ 2\ ]^T$
    - wartość funkcji celu: 1
  - rozwiązanie otrzymane metodą PC jest nieujemnym przybliżeniem rozwiązania układu wyjściowego

Programowanie celowe #37

- Rozwiązania przykładowych PC
  - znaleźć rozwiązanie zadania PC:

    min $|1x_1+2x_2-5|+|3x_1+2x_2-7|+|2x_1+1x_2-3|$

    p.o. $5x_1+7x_2 \leq 10$

    $x \geq 0,\ x_2 \geq 0$

    - rozwiązanie optymalne: $[\ 1.222...,\ 0.555...\ ]^T$
    - wartość funkcji celu: 4.888...