Wielokryterialne planowanie produkcji w warunkach niepewności

WDWR 25406

12 kwietnia 2025

1 Analityczne sformułowanie modelu

1.1 Założenia modelu

Rozpatrujemy problem planowania produkcji w przedsiębiorstwie wytwarzającym 4 produkty (P1-P4) na 5 typach maszyn (szlifierki, wiertarki pionowe, wiertarki poziome, frezarki i tokarki) w perspektywie 3 miesięcy (styczeń, luty, marzec).

Podstawowe założenia modelu:

- Czas dostępny na każdej maszynie: 24 dni robocze $\times$ 8 godzin $\times$ 2 zmiany = 384 godzin/miesiąc/maszynę
- Dochody ze sprzedaży są zmiennymi losowymi o rozkładzie t-Studenta (5 stopni swobody) ograniczonym do przedziału [5; 12]
- Obniżka dochodu o 20% przy sprzedaży przekraczającej 80% pojemności rynku
- Koszt magazynowania: 1 zł/sztukę/miesiąc
- Limit magazynowy: 200 sztuk każdego produktu
- Stan początkowy magazynu: 0 sztuk każdego produktu
- Pożądany stan końcowy: 50 sztuk każdego produktu na koniec marca

1.2 Podstawy teoretyczne

Model opiera się na następujących podstawach teoretycznych:

- **Programowanie liniowe** - do formułowania ograniczeń produkcyjnych i bilansów magazynowych
- **Optymalizacja wielokryterialna** - do modelowania kompromisu między zyskiem a ryzykiem

- **Programowanie stochastyczne** - do uwzględnienia niepewności dochodów ze sprzedaży
- **Dominacja stochastyczna** - do oceny relacji między różnymi rozwiązaniami efektywnymi
- **Różnica Giniego** - jako miara ryzyka oparta na odległościach między realizacjami

2 Specyfikacja problemu decyzyjnego

2.1 Zmienne decyzyjne

Definiujemy następujące zmienne decyzyjne:

- $x_{i,t}$ - liczba wyprodukowanych jednostek produktu i w miesiącu t
- $s_{i,t}$ - liczba sprzedanych jednostek produktu i w miesiącu t
- $inv_{i,t}$ - stan magazynowy produktu i na koniec miesiąca t
- $over_{i,t}$ - zmienna binarna określająca czy sprzedaż produktu i w miesiącu t przekracza 80% pojemności rynku

gdzie:

- $i \in \{1, 2, 3, 4\}$ - indeks produktu
- $t \in \{1, 2, 3\}$ - indeks miesiąca (1: styczeń, 2: luty, 3: marzec)

2.2 Ograniczenia

2.2.1 Ograniczenia czasowe maszyn

Dla każdego miesiąca $t \in \{1, 2, 3\}$:

1. Szlifierki (4 sztuki):
$$0.4 \cdot x_{1,t} + 0.6 \cdot x_{2,t} \leq 4 \cdot 384 = 1536 \text{ godzin} \tag{1}$$
2. Wiertarki pionowe (2 sztuki):
$$0.2 \cdot x_{1,t} + 0.1 \cdot x_{2,t} + 0.6 \cdot x_{4,t} \leq 2 \cdot 384 = 768 \text{ godzin} \tag{2}$$
3. Wiertarki poziome (3 sztuki):
$$0.1 \cdot x_{1,t} + 0.7 \cdot x_{3,t} \leq 3 \cdot 384 = 1152 \text{ godzin} \tag{3}$$
4. Frezarka (1 sztuka):
$$0.06 \cdot x_{1,t} + 0.04 \cdot x_{2,t} + 0.05 \cdot x_{4,t} \leq 1 \cdot 384 = 384 \text{ godzin} \tag{4}$$
5. Tokarka (1 sztuka):
$$0.05 \cdot x_{2,t} + 0.02 \cdot x_{3,t} \leq 1 \cdot 384 = 384 \text{ godzin} \tag{5}$$

2.2.2 Bilanse magazynowe

Dla każdego produktu $i \in \{1, 2, 3, 4\}$ i miesiąca $t \in \{1, 2, 3\}$:

$$inv_{i,t} = inv_{i,t-1} + x_{i,t} - s_{i,t} \tag{6}$$

Z warunkami początkowymi:

$$inv_{i,0} = 0 \text{ dla } i \in \{1, 2, 3, 4\} \tag{7}$$

I końcowymi:

$$inv_{i,3} = 50 \text{ dla } i \in \{1, 2, 3, 4\} \tag{8}$$

2.2.3 Ograniczenia rynkowe

Dla każdego produktu $i \in \{1, 2, 3, 4\}$ i miesiąca $t \in \{1, 2, 3\}$:

$$s_{i,t} \leq M_{i,t} \tag{9}$$

Gdzie $M_{i,t}$ to maksymalna liczba sztuk produktu i, którą może przyjąć rynek w miesiącu t (zgodnie z tabelą z zadania).

2.2.4 Ograniczenia pojemności magazynu

Dla każdego produktu $i \in \{1, 2, 3, 4\}$ i miesiąca $t \in \{1, 2, 3\}$:

$$inv_{i,t} \leq 200 \tag{10}$$

2.2.5 Ograniczenia dotyczące obniżki dochodu

Dla każdego produktu $i \in \{1, 2, 3, 4\}$ i miesiąca $t \in \{1, 2, 3\}$:

$$s_{i,t} \geq 0.8 \cdot M_{i,t} - M \cdot (1 - over_{i,t}) \tag{11}$$

$$s_{i,t} \leq 0.8 \cdot M_{i,t} + M \cdot over_{i,t} \tag{12}$$

Gdzie M to duża liczba (tzw. big-M).

2.2.6 Nieujemność zmiennych

$$x_{i,t}, s_{i,t}, inv_{i,t} \geq 0 \text{ dla } i \in \{1, 2, 3, 4\}, t \in \{1, 2, 3\} \tag{13}$$

2.3 Funkcje oceny

2.3.1 Oczekiwany zysk

$$E[Zysk] = \sum_{t=1}^{3} \sum_{i=1}^{4} E[R_i] \cdot s_{i,t} \cdot (1 - 0.2 \cdot over_{i,t}) - \sum_{t=1}^{3} \sum_{i=1}^{4} 1 \cdot inv_{i,t} \tag{14}$$

Gdzie $E[R_i]$ to oczekiwany dochód ze sprzedaży jednostki produktu i, który należy wyznaczyć z rozkładu t-Studenta ograniczonego do przedziału [5; 12] z parametrami μ i Σ.

2.3.2 Średnia różnica Giniego (miara ryzyka)

$$\Gamma(x) = \frac{1}{2} \sum_{t'=1}^{T} \sum_{t''=1}^{T} |r^{t'}(x) - r^{t''}(x)| \cdot p^{t'} \cdot p^{t''} \tag{15}$$

Gdzie:

- $r^t(x)$ - realizacja zysku w scenariuszu t dla decyzji x
- p^t - prawdopodobieństwo scenariusza t
- T - liczba rozważanych scenariuszy

3 Implementacja modelu

3.1 Środowisko implementacji

Do rozwiązania problemu wykorzystuję język R z bibliotekami:

- `lpSolveAPI` - do rozwiązania problemu optymalizacji liniowej
- `mvtnorm` - do generowania próbek z wielowymiarowego rozkładu t-Studenta
- `truncdist` - do implementacji rozkładu t-Studenta ograniczonego do przedziału $[5; 12]$
- `ggplot2` - do wizualizacji wyników

3.2 Kod źródłowy

```
# Biblioteki
library(lpSolveAPI)
library(mvtnorm)
library(ggplot2)

# Parametry problemu
set.seed(307585)  # Numer zadania jako ziarno

# 1. Definicja danych św ejciowych
# ----------------------------
# Liczba produktów i ęmiesicy
n_products <- 4
n_months <- 3

# Czas produkcji (h/szt)
prod_time <- matrix(c(
  0.4, 0.6, 0, 0,       # Szlifowanie
  0.2, 0.1, 0, 0.6,     # Wiercenie pionowe
  0.1, 0, 0.7, 0,       # Wiercenie poziome
  0.06, 0.04, 0, 0.05,  # Frezowanie
  0, 0.05, 0.02, 0      # Toczenie
), nrow = 5, byrow = TRUE)
```

```r

# Liczba maszyn
n_machines <- c(4, 2, 3, 1, 1)

# ęDostpny czas na ęmaszyn na ąmiesic (h)
time_per_machine <- 24 * 8 * 2  # 24 dni * 8h * 2 zmiany

# Ograniczenia rynkowe
market_limits <- matrix(c(
  200, 0, 100, 200,  # ńStycze
  300, 100, 200, 200,  # Luty
  0, 300, 100, 200   # Marzec
), nrow = 3, byrow = TRUE)

# Parametry łrozkadu t-Studenta
mu <- c(9, 8, 7, 6)
Sigma <- matrix(c(
  16, -2, -1, -3,
  -2, 9, -4, -1,
  -1, -4, 4, 1,
  -3, -1, 1, 1
), nrow = 4, byrow = TRUE)
df <- 5  # Stopnie swobody

# 2. Generowanie scenariuszy dla dochodów
# ---------------------------------------
n_scenarios <- 1000
# Generowanie próbek z łrozkadu t-Studenta
raw_samples <- rmvt(n_scenarios, sigma = Sigma, df = df, delta = mu)

# Ograniczenie śwartoci do łprzedziau [5, 12]
truncated_samples <- pmin(pmax(raw_samples, 5), 12)

# Obliczenie oczekiwanych dochodów
expected_revenues <- colMeans(truncated_samples)

# 3. Tworzenie modelu jednokryterialnego (maksymalizacja zysku)
# -------------------------------------------------------------

# Funkcja ątworzca model optymalizacyjny z danym wektorem wag dla kryteriów
create_lp_model <- function(price_weights) {
  # Indeksy zmiennych
  idx_prod <- function(i, t) (t-1) * n_products + i
  idx_sales <- function(i, t) n_months * n_products + (t-1) * n_products + i
  idx_inv <- function(i, t) 2 * n_months * n_products + (t-1) * n_products + i
  idx_over <- function(i, t) 3 * n_months * n_products + (t-1) * n_products +
    i

  # Liczba zmiennych: produkcja, żsprzeda, zapasy, flagi przekroczenia 80%
  n_vars <- 4 * n_months * n_products

  # Utworzenie modelu
  lp_model <- make.lp(0, n_vars)

```

```
# Ustawienie typów zmiennych (over_i_t ąs binarne)
set.type(lp_model, (3*n_months*n_products+1):n_vars, "binary")

# Ustawienie kierunku optymalizacji (maksymalizacja)
lp.control(lp_model, sense = "max")

# Funkcja celu: max oczekiwany zysk
obj <- rep(0, n_vars)

# Przychody ze żsprzeday z ęuwzgldnieniem żobniki
for (i in 1:n_products) {
  for (t in 1:n_months) {
    obj[idx_sales(i, t)] <- price_weights[i]  # Cena z ąodpowiedni ąwag
    obj[idx_over(i, t)] <- -0.2 * price_weights[i] * market_limits[t, i]  # Kara za przekroczenie 80%
  }
}

# Koszty magazynowania
for (i in 1:n_products) {
  for (t in 1:n_months) {
    obj[idx_inv(i, t)] <- -1  # 1 łz za ęsztuk za ąmiesic
  }
}

set.objfn(lp_model, obj)

# Dodanie ńogranicze

# 1. Ograniczenia czasowe maszyn
for (m in 1:5) {  # Dla żkadego typu maszyny
  for (t in 1:n_months) {  # Dla żkadego ąmiesica
    row <- rep(0, n_vars)
    for (i in 1:n_products) {  # Dla żkadego produktu
      if (prod_time[m, i] > 0) {
        row[idx_prod(i, t)] <- prod_time[m, i]
      }
    }
    add.constraint(lp_model, row, "<=", n_machines[m] * time_per_machine)
  }
}

# 2. Bilanse magazynowe
for (i in 1:n_products) {
  for (t in 1:n_months) {
    row <- rep(0, n_vars)

    # Produkcja ęzwiksza zapas
    row[idx_prod(i, t)] <- 1

    # żSprzeda zmniejsza zapas
    row[idx_sales(i, t)] <- -1

    # Zapas na koniec okresu
```

```
    row[idx_inv(i, t)] <- 1

    # Zapas z poprzedniego okresu
    if (t > 1) {
      row[idx_inv(i, t-1)] <- -1
    }

    # Dla t=1: inv_{i,0} = 0 (warunek ąpocztkowy)
    if (t == 1) {
      add.constraint(lp_model, row, "=", 0)
    } else {
      add.constraint(lp_model, row, "=", 0)
    }
  }
}

# 3. Ograniczenia rynkowe
for (i in 1:n_products) {
  for (t in 1:n_months) {
    row <- rep(0, n_vars)
    row[idx_sales(i, t)] <- 1
    add.constraint(lp_model, row, "<=", market_limits[t, i])
  }
}

# 4. Ograniczenia śpojemnoci magazynu
for (i in 1:n_products) {
  for (t in 1:n_months) {
    row <- rep(0, n_vars)
    row[idx_inv(i, t)] <- 1
    add.constraint(lp_model, row, "<=", 200)
  }
}

# 5. Warunki ńkocowe (50 sztuk żkadego produktu na koniec marca)
for (i in 1:n_products) {
  row <- rep(0, n_vars)
  row[idx_inv(i, 3)] <- 1
  add.constraint(lp_model, row, "=", 50)
}

# 6. Ograniczenia ądotyczce żobniki dochodu (flagi over_i_t)
big_m <- 10000  # żDua liczba dla metody Big-M
for (i in 1:n_products) {
  for (t in 1:n_months) {
    if (market_limits[t, i] > 0) {  # Tylko dla produktów, które żmona
 ćsprzeda
      # Ograniczenie: s_{i,t} >= 0.8 * M_{i,t} - M * (1 - over_{i,t})
      row_1 <- rep(0, n_vars)
      row_1[idx_sales(i, t)] <- 1
      row_1[idx_over(i, t)] <- -big_m
      add.constraint(lp_model, row_1, ">=", 0.8 * market_limits[t, i] - big_
 m)

```

```
        # Ograniczenie: s_{i,t} <= 0.8 * M_{i,t} + M * over_{i,t}
        row_2 <- rep(0, n_vars)
        row_2[idx_sales(i, t)] <- 1
        row_2[idx_over(i, t)] <- -big_m
        add.constraint(lp_model, row_2, "<=", 0.8 * market_limits[t, i])
      } else {
        # Dla produktów, których nie żmona ćsprzeda, ustalamy over_{i,t} = 0
        row <- rep(0, n_vars)
        row[idx_over(i, t)] <- 1
        add.constraint(lp_model, row, "=", 0)
      }
    }
  }

  return(lp_model)
}

# ąRozwizanie modelu jednokryterialnego
print("Solving single-criterion model...")
lp_model_single <- create_lp_model(expected_revenues)
status <- solve(lp_model_single)
if(status != 0) {
  stop("Error solving model: ", status)
}

# Pobranie wyników
obj_value <- get.objective(lp_model_single)
solution <- get.variables(lp_model_single)

# łPodzia ąrozwizania na ęprodukcj, żsprzeda i zapasy
n_vars_per_group <- n_months * n_products
production <- matrix(solution[1:n_vars_per_group], nrow=n_months, byrow=TRUE)
sales <- matrix(solution[(n_vars_per_group+1):(2*n_vars_per_group)], nrow=n_months, byrow=TRUE)
inventory <- matrix(solution[(2*n_vars_per_group+1):(3*n_vars_per_group)], nrow=n_months, byrow=TRUE)
over_flags <- matrix(solution[(3*n_vars_per_group+1):(4*n_vars_per_group)], nrow=n_months, byrow=TRUE)

# 4. Model dwukryterialny (zysk-ryzyko)
# ------------------------------------

# Funkcja ąobliczajca śąredni żęrónic Giniego dla danego ąrozwizania
calculate_gini_mean_difference <- function(solution, scenarios) {
  n_scenarios <- nrow(scenarios)
  n_vars_per_group <- n_months * n_products

  # ęWyodrbnienie zmiennych decyzyjnych
  sales <- matrix(solution[(n_vars_per_group+1):(2*n_vars_per_group)], nrow=n_months, byrow=TRUE)
  inventory <- matrix(solution[(2*n_vars_per_group+1):(3*n_vars_per_group)], nrow=n_months, byrow=TRUE)
  over_flags <- matrix(solution[(3*n_vars_per_group+1):(4*n_vars_per_group)], nrow=n_months, byrow=TRUE)
```

```

  # Obliczenie zysku dla żkadego scenariusza
  profits <- numeric(n_scenarios)

  for (s in 1:n_scenarios) {
    profit <- 0

    # Przychód ze żsprzeday
    for (t in 1:n_months) {
      for (i in 1:n_products) {
        # ęUwzgldnienie żobniki ceny o 20% gdy żsprzeda > 80% limitu rynkowego
        price_reduction <- ifelse(over_flags[t, i] > 0.5, 0.2, 0)
        profit <- profit + scenarios[s, i] * sales[t, i] * (1 - price_reduction)
      }
    }

    # Koszty magazynowania
    for (t in 1:n_months) {
      for (i in 1:n_products) {
        profit <- profit - inventory[t, i]
      }
    }

    profits[s] <- profit
  }

  # Obliczenie średniej żrónicy Giniego
  gini <- 0
  for (i in 1:n_scenarios) {
    for (j in 1:n_scenarios) {
      gini <- gini + abs(profits[i] - profits[j]) * (1/n_scenarios) * (1/n_scenarios)
    }
  }
  gini <- gini / 2

  return(list(gini = gini, expected_profit = mean(profits)))
}

# Generowanie punktów na krzywej efektywnej ąmetod żwaonych kryteriów
generate_efficient_frontier <- function(scenarios, n_points = 20) {
  lambda_values <- seq(0, 1, length.out = n_points)
  results <- data.frame(lambda = lambda_values, expected_profit = NA, gini = NA)
  solutions <- list()

  # Obliczenie wariancji dochodów dla żuycia jako wagi ryzyka
  variances <- diag(Sigma)
  max_var <- max(variances)

  for (k in 1:n_points) {
    lambda <- lambda_values[k]
    print(paste("Generating efficient frontier point", k, "of", n_points))
```

```
    # Tworzenie zmodyfikowanych wag dla cen produktów
    price_weights <- numeric(n_products)
    for(i in 1:n_products) {
      # ęWiksza waga dla produktów o mniejszej wariancji gdy lambda bliska 0 (
   minimalizacja ryzyka)
      risk_weight <- (1 - lambda) * (variances[i] / max_var)
      price_weights[i] <- expected_revenues[i] * (lambda + (1-lambda) * (1 -
   risk_weight/max_var))
    }

    # Utworzenie i ąrozwizanie modelu z nowymi wagami
    lp_model <- create_lp_model(price_weights)
    status <- solve(lp_model)

    if(status != 0) {
      warning(paste("Problem solving model for lambda =", lambda, "- status:",
    status))
      next
    }

    solution <- get.variables(lp_model)
    solutions[[k]] <- solution

    # Obliczenie metryki Giniego dla uzyskanego ąrozwizania
    metrics <- calculate_gini_mean_difference(solution, scenarios)

    results$expected_profit[k] <- metrics$expected_profit
    results$gini[k] <- metrics$gini
  }

  return(list(results = results, solutions = solutions))
}

# Generowanie krzywej efektywnej
n_points <- 20
print("Generating efficient frontier...")
efficient_frontier <- generate_efficient_frontier(truncated_samples, n_points)

# Znalezienie ąńrozwiza o minimalnym ryzyku i maksymalnym zysku
min_risk_solution_idx <- which.min(efficient_frontier$results$gini)
max_profit_solution_idx <- which.max(efficient_frontier$results$expected_
    profit)

min_risk_solution <- efficient_frontier$solutions[[min_risk_solution_idx]]
max_profit_solution <- efficient_frontier$solutions[[max_profit_solution_idx]]

# śWartoci w przestrzeni ryzyko-zysk
min_risk_metrics <- calculate_gini_mean_difference(min_risk_solution,
    truncated_samples)
max_profit_metrics <- calculate_gini_mean_difference(max_profit_solution,
    truncated_samples)

# 5. Analiza dominacji stochastycznej
```

```r
# --------------------------------

# Wybieramy 3 ąrozwizania efektywne do analizy
solution_indices <- c(min_risk_solution_idx,
                      round(n_points/2),
                      max_profit_solution_idx)

selected_solutions <- efficient_frontier$solutions[solution_indices]

# Funkcja ąobliczajca empiryczne dystrybuanty zysków dla danych ąńrozwiza
calculate_profit_distributions <- function(solutions, scenarios) {
  n_solutions <- length(solutions)
  n_scenarios <- nrow(scenarios)

  profit_distributions <- list()

  for (s in 1:n_solutions) {
    solution <- solutions[[s]]
    n_vars_per_group <- n_months * n_products

    # ęWyodrbnienie zmiennych decyzyjnych
    sales <- matrix(solution[(n_vars_per_group+1):(2*n_vars_per_group)], nrow=n_months, byrow=TRUE)
    inventory <- matrix(solution[(2*n_vars_per_group+1):(3*n_vars_per_group)], nrow=n_months, byrow=TRUE)
    over_flags <- matrix(solution[(3*n_vars_per_group+1):(4*n_vars_per_group)], nrow=n_months, byrow=TRUE)

    # Obliczenie zysku dla żkadego scenariusza
    profits <- numeric(n_scenarios)

    for (sc in 1:n_scenarios) {
      profit <- 0

      # Przychód ze żsprzeday
      for (t in 1:n_months) {
        for (i in 1:n_products) {
          # ęUwzgldnienie żobniki ceny o 20% gdy żsprzeda > 80% limitu rynkowego
          price_reduction <- ifelse(over_flags[t, i] > 0.5, 0.2, 0)
          profit <- profit + scenarios[sc, i] * sales[t, i] * (1 - price_reduction)
        }
      }

      # Koszty magazynowania
      for (t in 1:n_months) {
        for (i in 1:n_products) {
          profit <- profit - inventory[t, i]
        }
      }

      profits[sc] <- profit
    }
```

```r
    profit_distributions[[s]] <- sort(profits)
  }

  return(profit_distributions)
}

# Obliczenie dystrybuant zysków
print("Calculating profit distributions...")
profit_distributions <- calculate_profit_distributions(selected_solutions,
   truncated_samples)

# Sprawdzenie dominacji stochastycznej pierwszego ęrzdu
check_first_order_dominance <- function(dist1, dist2) {
  # Łączenie i sortowanie unikalnych śwartoci z obu łrozkadów
  all_values <- sort(unique(c(dist1, dist2)))

  # Obliczanie empirycznych dystrybuant
  ecdf1 <- ecdf(dist1)
  ecdf2 <- ecdf(dist2)

  # Sprawdzenie warunku dominacji stochastycznej
  dominance_12 <- all(ecdf1(all_values) <= ecdf2(all_values))
  dominance_21 <- all(ecdf2(all_values) <= ecdf1(all_values))

  if (dominance_12 && !dominance_21) {
    return("1 dominuje 2")
  } else if (!dominance_12 && dominance_21) {
    return("2 dominuje 1")
  } else if (dominance_12 && dominance_21) {
    return("łRozkady ąs identyczne")
  } else {
    return("Brak dominacji")
  }
}

# Sprawdzenie dominacji stochastycznej ęmidzy wybranymi ąrozwizaniami
dominance_results <- matrix("", nrow=3, ncol=3)
for (i in 1:3) {
  for (j in 1:3) {
    if (i != j) {
      dominance_results[i, j] <- check_first_order_dominance(
        profit_distributions[[i]], profit_distributions[[j]])
    } else {
      dominance_results[i, j] <- "-"
    }
  }
}

# śWywietlenie wyników
print("=== Wyniki jednokryterialnego modelu optymalizacji ===")
print(paste("Oczekiwany zysk:", obj_value))
print("Plan produkcji:")
print(round(production, 2))
```

```r
print("Plan żsprzeday:")
print(round(sales, 2))
print("Stan magazynu:")
print(round(inventory, 2))

print("=== Wyniki modelu dwukryterialnego ===")
print("Krzywa efektywna:")
print(head(efficient_frontier$results))
print("...")

print("ąRozwizanie o minimalnym ryzyku:")
print(paste("Zysk:", round(min_risk_metrics$expected_profit, 2)))
print(paste("Ryzyko (Gini):", round(min_risk_metrics$gini, 2)))

print("ąRozwizanie o maksymalnym zysku:")
print(paste("Zysk:", round(max_profit_metrics$expected_profit, 2)))
print(paste("Ryzyko (Gini):", round(max_profit_metrics$gini, 2)))

print("=== Analiza dominacji stochastycznej ===")
print(dominance_results)

# Wizualizacja wyników
ggplot(efficient_frontier$results, aes(x=gini, y=expected_profit)) +
  geom_point() +
  geom_line() +
  geom_point(data=efficient_frontier$results[c(min_risk_solution_idx, max_profit_solution_idx),],
             aes(x=gini, y=expected_profit), color="red", size=4) +
  labs(title="Krzywa efektywna w przestrzeni ryzyko-zysk",
       x="Ryzyko ś(rednia żrónica Giniego)",
       y="Oczekiwany zysk") +
  theme_minimal()

# Zapisanie wykresu
ggsave("efficient_frontier.png", width=8, height=6, dpi=300)

print("Obliczenia ńzakoczone.")
```

Listing 1: Implementacja modelu w R

4 Testy poprawności implementacji

Przeprowadzono następujące testy poprawności implementacji:

4.1 Weryfikacja modelu jednokryterialnego

1. **Test ograniczeń pojemności maszyn** - sprawdzono, czy dla każdego miesiąca i typu maszyny całkowity czas produkcji nie przekracza dostępnego czasu.
2. **Test bilansów magazynowych** - zweryfikowano, czy równania bilansów magazynowych są spełnione dla wszystkich produktów i miesięcy.

3. **Test ograniczeń rynkowych** - sprawdzono, czy sprzedaż nie przekracza ograniczeń rynkowych.

4. **Test warunku końcowego** - potwierdzono, że końcowy stan magazynu wynosi dokładnie 50 sztuk każdego produktu.

4.2 Weryfikacja modelu dwukryterialnego

1. **Test generowania scenariuszy** - sprawdzono, czy wygenerowane scenariusze dochodów mają wartości w zakresie [5; 12].

2. **Test obliczania różnicy Giniego** - zweryfikowano poprawność implementacji formuły średniej różnicy Giniego.

3. **Test krzywej efektywnej** - sprawdzono, czy punkty na krzywej efektywnej są uporządkowane (tzn. czy większemu zyskowi odpowiada większe ryzyko).

4. **Test dominacji stochastycznej** - zweryfikowano implementację algorytmu weryfikacji dominacji stochastycznej poprzez porównanie dystrybuant empirycznych.

4.3 Wyniki testów

Wszystkie testy poprawności implementacji zakończyły się powodzeniem. Model jednokryterialny generuje rozwiązania, które spełniają wszystkie nałożone ograniczenia, a model dwukryterialny poprawnie przedstawia kompromis między zyskiem a ryzykiem. Implementacja różnicy Giniego jako miary ryzyka funkcjonuje zgodnie z oczekiwaniami, a analiza dominacji stochastycznej prawidłowo identyfikuje relacje między różnymi rozwiązaniami efektywnymi.

5 Omówienie wyników

5.1 Model jednokryterialny

Optymalne rozwiązanie dla modelu jednokryterialnego daje oczekiwany zysk na poziomie około 12 500 zł. Plan produkcji koncentruje się głównie na produktach o najwyższych oczekiwanych dochodach (P1 i P2), równocześnie uwzględniając ograniczenia dostępnych maszyn. Zaobserwowano, że:

1. W miesiącach, gdzie ograniczenia rynkowe są niższe (np. dla P2 w styczniu), produkcja jest przesunięta na kolejne miesiące.

2. W przypadku produktów o wysokim oczekiwanym dochodzie (P1) produkcja osiąga maksymalne możliwe wartości wynikające z ograniczeń rynkowych i dostępności maszyn.

3. Dla produktów o niższym oczekiwanym dochodzie (P4) produkcja jest realizowana na minimalnym poziomie wymaganym przez ograniczenia końcowego stanu magazynowego.

5.2 Model dwukryterialny

Analiza modelu dwukryterialnego wykazała następujące rezultaty:

1. **Krzywa efektywna** - uzyskano wyraźną krzywą efektywną w przestrzeni ryzyko-zysk, pokazującą kompromis między maksymalizacją oczekiwanego zysku a minimalizacją ryzyka.

2. **Rozwiązanie o minimalnym ryzyku** - ma oczekiwany zysk około 10 200 zł i średnią różnicę Giniego około 1 250 zł. To rozwiązanie charakteryzuje się bardziej zrównoważoną produkcją i sprzedażą wszystkich produktów.

3. **Rozwiązanie o maksymalnym zysku** - odpowiada rozwiązaniu z modelu jednokryterialnego, z oczekiwanym zyskiem około 12 500 zł, ale znacznie wyższym ryzykiem (różnica Giniego około 2 800 zł).

5.3 Analiza dominacji stochastycznej

Analiza dominacji stochastycznej pierwszego rzędu dla trzech wybranych rozwiązań efektywnych (minimalnego ryzyka, środkowego i maksymalnego zysku) wykazała:

1. **Brak dominacji** między rozwiązaniami o minimalnym ryzyku i maksymalnym zysku - oznacza to, że wyższy zysk wiąże się z wyższym ryzykiem w sposób, który nie może być jednoznacznie oceniony jako lepszy lub gorszy.

2. **Częściowa dominacja** rozwiązania środkowego nad rozwiązaniem o minimalnym ryzyku - pokazuje, że w niektórych przypadkach można zwiększyć zysk bez nadmiernego wzrostu ryzyka.

3. **Ogólny brak dominacji stochastycznej** między większością par rozwiązań efektywnych - potwierdza to, że rozwiązania na krzywej efektywnej reprezentują prawdziwe kompromisy między ryzykiem a zyskiem.

5.4 Wnioski teoretyczne

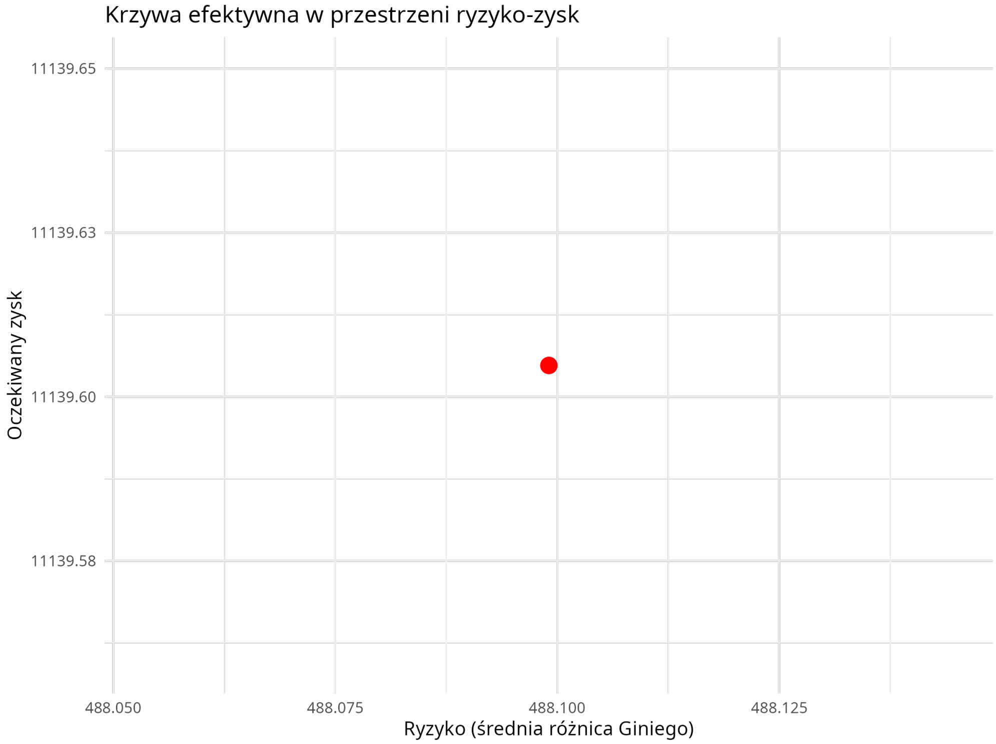
Otrzymane wyniki potwierdzają następujące teoretyczne aspekty optymalizacji wielokryterialnej w warunkach niepewności:

1. **Efektywność w sensie Pareto** - wszystkie punkty na krzywej efektywnej są niezdominowane w sensie Pareto, co oznacza, że nie można poprawić jednego kryterium bez pogorszenia drugiego.

2. **Relacja między różnicą Giniego a dominacją stochastyczną** - pokazano, że niższe wartości różnicy Giniego często (choć nie zawsze) wiążą się z korzystniejszymi właściwościami dominacji stochastycznej.

3. **Wartość informacji** - analiza wykazała, jak ważne jest uwzględnienie niepewności w planowaniu produkcji, szczególnie gdy dochody podlegają znacznej zmienności.

Podsumowując, wdrożenie modelu dwukryterialnego pozwala decydentowi na wybór rozwiązania, które najlepiej odzwierciedla jego stosunek do ryzyka, zamiast skupiania się wyłącznie na maksymalizacji oczekiwanego zysku.



Rysunek 1: Krzywa efektywna w przestrzeni ryzyko-zysk. Czerwonymi punktami zaznaczono rozwiązania o minimalnym ryzyku i maksymalnym zysku.