Discrete Optimization - Assignment II

Student Name: Luke Renton

Student number: 2540440

Due date: 24 October 2025

Question 1 - Fiduccia & Mattheyses (FM) Algorithm with Penalty Function

Question 1 Code Implementation

```
# ============================================================
# Assignment II - COMS4050/7057 (Discrete Optimization)
# Fiduccia & Mattheyses (FM) Algorithm with Penalty Function
# Due Date: 24 October 2025
# Student Name: Luke Renton
# Student number: 2540440
# ============================================================

values = [6, 8, 3, 4, 5, 9, 11, 12, 6, 8, 13, 15, 16, 13, 9, 25]
weights = [3, 5, 4, 7, 4, 10, 3, 6, 8, 14, 4, 9, 10, 11, 17, 12]
W = 25
n = len(values)
R = 200  # (penalty strength)

# Initial solution: xi = 1 if i is odd (1-indexed), else 0
x = [1 if (i + 1) % 2 == 1 else 0 for i in range(n)]

def total_weight(x): return sum(w * xi for w, xi in zip(weights, x))
def total_value(x): return sum(v * xi for v, xi in zip(values, x))

def phi(x):
    """Penalty component: excess weight over capacity."""
    return max(0, total_weight(x) - W)

def fitness(x):
    """Fitness with R=200 penalty term."""
    return total_value(x) - R * phi(x)

x_best = x[:]
f_best = fitness(x)
flag = 1
Pass = 1

print("============================================================")
print("FIDUCCIA & MATTHEYSES (FM) LOCAL SEARCH WITH PENALTY FUNCTION")
```

```
print("================================================================")
print(f"Initial solution: x = {x}")
print(f"Initial weight = {total_weight(x)}, value = {total_value(x)}, fitness =
{f_best:.2f}")
print("----------------------------------------------------------------")

while flag == 1:
    flag = 0
    Epoch = 0
    F = list(range(n))
    L = []
    x_epoch_best = x[:]
    f_epoch_best = fitness(x)

    print(f"\n=== PASS {Pass} START ===")

    while F:
        Epoch += 1
        best_gain = float('-inf')
        best_index = None

        for j in F:
            x_try = x[:]
            x_try[j] = 1 - x_try[j]
            gain = fitness(x_try) - fitness(x)
            if gain > best_gain:
                best_gain = gain
                best_index = j

        if best_index is None:
            break

        # Apply best flip
        x[best_index] = 1 - x[best_index]
        F.remove(best_index)
        L.append(best_index)
        f_curr = fitness(x)
        w_curr = total_weight(x)

        print(f"Epoch {Epoch:2d}: flipped index {best_index:2d}, "
              f"weight = {w_curr}, fitness = {f_curr:.2f}, gain =
{best_gain:+.2f}")

        if f_curr > f_epoch_best:
            f_epoch_best = f_curr
            x_epoch_best = x[:]

    # End of pass
    if f_epoch_best > f_best:
        f_best = f_epoch_best
        x_best = x_epoch_best[:]
        x = x_best[:]
        flag = 1
        Pass += 1
```

```
        print(f"→ Improved global best: fitness = {f_best:.2f}, weight =
{total_weight(x_best)}")
    else:
        print("→ No further improvement. Terminating search.")
        break

print("\n================================================================")
print("FINAL RESULTS")
print("================================================================")
print(f"Best solution x* = {x_best}")
print(f"Total weight = {total_weight(x_best)}")
print(f"Total value  = {total_value(x_best)}")
print(f"Final penalized fitness = {fitness(x_best):.2f}")
print("================================================================")
```

Question 1 outputs

> For brevity, only key steps and improvements are shown in the output below. The full algorithm executes each epoch internally, but repetitive flip-level details were omitted for clarity.

```
================================================================
FIDUCCIA & MATTHEYSES (FM) LOCAL SEARCH WITH PENALTY FUNCTION
================================================================
Initial solution: x = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]
Initial weight = 53, value = 69, fitness = -5531.00
----------------------------------------------------------------

=== PASS 1 START ===
Epoch  1: flipped index 14, weight = 36, fitness = -2140.00, gain = +3391.00
Epoch  2: flipped index 12, weight = 26, fitness = -156.00, gain = +1984.00
Epoch  3: flipped index  2, weight = 22, fitness = 41.00, gain = +197.00
Epoch  4: flipped index  4, weight = 18, fitness = 36.00, gain = -5.00
Epoch  5: flipped index  7, weight = 24, fitness = 48.00, gain = +12.00
Epoch  6: flipped index  0, weight = 21, fitness = 42.00, gain = -6.00
Epoch  7: flipped index  8, weight = 13, fitness = 36.00, gain = -6.00
Epoch  8: flipped index 15, weight = 25, fitness = 61.00, gain = +25.00
Epoch  9: flipped index  6, weight = 22, fitness = 50.00, gain = -11.00
Epoch 10: flipped index 10, weight = 18, fitness = 37.00, gain = -13.00
Epoch 11: flipped index  1, weight = 23, fitness = 45.00, gain = +8.00
Epoch 12: flipped index  3, weight = 30, fitness = -951.00, gain = -996.00
Epoch 13: flipped index 11, weight = 39, fitness = -2736.00, gain = -1785.00
Epoch 14: flipped index  5, weight = 49, fitness = -4727.00, gain = -1991.00
Epoch 15: flipped index 13, weight = 60, fitness = -6914.00, gain = -2187.00
Epoch 16: flipped index  9, weight = 74, fitness = -9706.00, gain = -2792.00
→ Improved global best: fitness = 61.00, weight = 25

=== PASS 2 START ===
Epoch  1: flipped index  6, weight = 22, fitness = 50.00, gain = -11.00
Epoch  2: flipped index  0, weight = 25, fitness = 56.00, gain = +6.00
Epoch  3: flipped index  7, weight = 19, fitness = 44.00, gain = -12.00
Epoch  4: flipped index  1, weight = 24, fitness = 52.00, gain = +8.00
```

```
Epoch  5: flipped index 10, weight = 20, fitness = 39.00, gain = -13.00
Epoch  6: flipped index  4, weight = 24, fitness = 44.00, gain = +5.00
Epoch  7: flipped index 15, weight = 12, fitness = 19.00, gain = -25.00
Epoch  8: flipped index 12, weight = 22, fitness = 35.00, gain = +16.00
Epoch  9: flipped index  2, weight = 26, fitness = -162.00, gain = -197.00
Epoch 10: flipped index  3, weight = 33, fitness = -1558.00, gain = -1396.00
Epoch 11: flipped index  8, weight = 41, fitness = -3152.00, gain = -1594.00
Epoch 12: flipped index 11, weight = 50, fitness = -4937.00, gain = -1785.00
Epoch 13: flipped index  5, weight = 60, fitness = -6928.00, gain = -1991.00
Epoch 14: flipped index 13, weight = 71, fitness = -9115.00, gain = -2187.00
Epoch 15: flipped index  9, weight = 85, fitness = -11907.00, gain = -2792.00
Epoch 16: flipped index 14, weight = 102, fitness = -15298.00, gain = -3391.00
→ No further improvement. Terminating search.

================================================================
FINAL RESULTS
================================================================
Best solution x* = [0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1]
Total weight = 25
Total value  = 61
Final penalized fitness = 61.00
================================================================
```