NGRER Analysis Model Formulation

Table of Contents

NGRER Analysis Mathematical Model Formulation

Mixed-Integer Linear Programming Model

Decision Variables:

- $\boldsymbol{x}_{\ (c,u,l,e,d)}$ = quantity of $\boldsymbol{LIN\ l}$ at modernization level $\boldsymbol{e}$ allocated to unit $\boldsymbol{u}$ in component $\boldsymbol{c}$ for year $\boldsymbol{d}$
- $\boldsymbol{s}_{\ (c,u,l,e,d)}$ = quantity of substitutions from higher modernization levels
- $\boldsymbol{h}_{\ (c,u,l,e,d)}$ = shortage quantity (unmet requirements)
- $\boldsymbol{t}_{\ (c^1,c^2,l,e,d)}$ = inter-component transfers from component $\boldsymbol{c^1}$ to $\boldsymbol{c^2}$

Indexes:

- $d\ (dates) \in (date_{min} \dots date_{max})$
- $c\ (compos) \in (1, 2, 3, 6)$
- $u\ (units) \in (uic_1 \dots uic_m)$
- $l\ (lins) \in (lin_1 \dots lin_n)$
- $s\ (sublins) \in (sublin_1 \dots sublin_n)$

Objective Function:

$$\boldsymbol{min} \sum_{\boldsymbol{c,u,l,e,d}} \boldsymbol{DARPL[c,u]} \times \boldsymbol{h_{c,u,l,e,d}} + \sum_{\boldsymbol{transfers}} \boldsymbol{TransferPenalty} \times \boldsymbol{t_{c^1,c^2,l,e,d}}$$

Critical Constraints:

1. **Inventory Conservation:**

$$\sum_{\boldsymbol{u,e}} \boldsymbol{x_{c,u,l,e,d}} + \sum_{\boldsymbol{c^2}} \boldsymbol{t_{c,c^2,l,e,d}} \leq \boldsymbol{I_{c,l,e,d}} + \sum_{\boldsymbol{c^1}} \boldsymbol{t_{c^1,c,l,e,d}}$$

This constraint ensures that total allocations plus outbound transfers cannot exceed available inventory plus inbound transfers.

2. **Requirement Satisfaction:**

$$\boldsymbol{x_{c,u,l,e,d}} + \boldsymbol{s_{c,u,l,e,d}} + \boldsymbol{h_{c,u,l,e,d}} = \boldsymbol{R_{c,u,l,e,d}}$$

3. **Substitution Constraints:**

$$\boldsymbol{s_{c,u,l,e,d}} \leq \sum_{\boldsymbol{l' \in Sub(l)}} \boldsymbol{x_{c,u,l',e',d}}$$

$$\boldsymbol{where\ Sub(l) represents\ valid\ substitutes\ for\ LIN\ l}$$

4. **Non-negativity:**

$$\boldsymbol{x_{c,u,l,e,d}}, \boldsymbol{s_{c,u,l,e,d}}, \boldsymbol{h_{c,u,l,e,d}}, \boldsymbol{t_{c^1,c^2,l,e,d}} \geq \boldsymbol{0}$$

5. **Integer Constraints:** All decision variables must be non-negative integers representing discrete equipment units.

Business Rules & Assumptions:

Summary of Critical Business Assumptions

- **Equipment Interchangeability:** Substitution relationships accurately reflect operational capability equivalence
- **Priority Accuracy:** DARPL values correctly represent relative unit importance
- **Cost Uniformity:** Transfer costs are consistent across all component pairs and geographic locations
- **Perfect Information:** All data sources provide complete and accurate equipment status
- **Implementation Feasibility:** Optimal solutions are practically implementable within Army constraints
- **Stable Requirements:** Future equipment needs can be accurately projected across 7-year horizon
- **Modernization Value:** Higher modernization levels provide proportionally superior capability
- **Resource Fungibility:** Equipment can be freely moved between authorized positions

1. **Equipment Readiness Code (ERC) Business Rules**
   - **ERC Hierarchy Business Rules:**
     - **ERC P (Primary):** Equipment essential for primary mission accomplishment receives highest optimization priority
     - **ERC A (Augmentation):** Equipment supporting expanded capabilities receives moderate priority
     - **ERC B/C:** Training and commercial equipment receives minimal priority and is generally filtered out
     - **Rule:** ERC P shortages must be filled before any ERC A assignments are made
     - **Assumption:** ERC categories accurately reflect mission criticality across all unit types
     - ***SAS:***

```
%let p_pri = 10000000000;  /* ERC P (Primary) - Mission Critical */
%let a_pri = 500;          /* ERC A (Augmentation) - Secondary */
%let b_pri = 100;          /* ERC B (Background) - Minimal */
%let c_pri = 10;           /* ERC C (Commercial) - Lowest */
```

   - **Fill Target Business Rules**
     - **Rule:** All units should achieve 100% equipment fill for their authorized requirements
     - **Assumption:** 100% fill is operationally necessary and achievable across the enterprise
     - **Constraint:** Ceiling function ensures no partial equipment assignments
     - ***SAS:***

```
%let fill_target = 1.0;  /* 100% fill requirement */
```

2. **Component Transfer Business Rules**
   - **Transfer Business Rules:**
     - **Component Structure:**
       - **Component 1 (AC):** Active Component
       - **Component 2 (ARNG):** Army National Guard
       - **Component 3 (USAR):** Army Reserve
       - **Component 6 (APS):** Army Prepositioned Stocks
     - **Bidirectional Transfers:** All components can send/receive equipment from all other components
     - **Transfer Penalty:** 150 penalty per unit transferred to discourage unnecessary movement
     - **Rule:** Transfers only occur when benefit exceeds movement cost
     - **Assumptions:**
       - Administrative and logistical transfer costs are uniform across all component pairs
       - All equipment is physically transferable between components

- **SAS:**

  ```
  %let valid_transfer = 1;   /* Enable transfers */
  %let trans_pen = 150;      /* Transfer penalty per unit */
  ```

- **Transfer Timing Constraints**
  - **Rule:** Equipment cascading (transfers from higher to lower priority units) restricted in early years
  - **Assumption:** Newer equipment should remain with high-priority units initially
  - **SAS:**

    ```
    %if %eval(&CASC_YEAR.) %then %do;
        /* Cascading restricted before specified year */
        for{<d,c,f,u,l> in ERC_P_XFER_SET} if d < &CASC_YEAR. then fix
    erc_p_xfer[d,c,f,u,l] = 0;
    %end;
    ```

3. **Modernization Level Business Rules**

- **Modernization Level Hierarchy**
  - **Rules:**
    - Modernization Level 5 (newest) equipment should be deployed, not stored as excess
    - Higher modernization levels receive exponentially higher excess penalties
    - Lower modernization levels (1-2) are acceptable for excess storage
  - **Assumptions:**
    - Newer equipment provides superior operational capability
    - Storing modern equipment represents significant opportunity cost
  - **SAS:**

    ```
    %let mod_5_e_pen = 999;    /* Newest equipment excess penalty */
    %let mod_4_e_pen = 499;    /* High excess penalty */
    %let mod_3_e_pen = 200;    /* Moderate excess penalty */
    %let mod_2_e_pen = 10;     /* Low excess penalty */
    %let mod_1_e_pen = 5;      /* Minimal excess penalty */
    ```

- **NGRER-Specific Modernization Rules**
  - **Rule:** NGRER analysis focuses only on Modernization Level 3+ equipment
  - **Assumption:** Legacy equipment (ML1-2) represents divest-capable inventory
  - **Business Rationale:** Focus resources on equipment worth retaining/modernizing
  - **SAS:**

    ```
    %if %eval(&NGRER_RUN_TOGGLE = 1) %then %do;
        /* Restrict assignment of ML1 and ML2 inventory */
        for {<d,c,u,l> in ERC_P_ASSIGN_SET} if mod_level[l] <3 then fix erc_P_assign[d,
    c, u, l] = 0;
    %end;
    ```

4. **Substitution Business Rules**

- **Substitution Source Hierarchy**
  - **Rules:**
    - SB 700-20 Appendix H substitutions have highest authority and priority
    - LMDB "replaces" relationships have moderate authority
    - LMDB "replaced by" relationships have lower authority
    - In-lieu-of substitutions (source ≥7) are prohibited
  - **Assumption:** Official Army publications override automated LMDB relationships
  - **SAS:**

    ```
    /* Substitution source priority ranking */
    if source = '1-SB700-20' then priority = 1;       /* Highest authority */
    else if source = '3-REPLACES' then priority = 3;  /* LMDB replaces */
    else if source = '4-REPLACED' then priority = 4;  /* LMDB replaced by */
    else if source >= 7 then /* In-lieu-of substitutions */ fix to 0;
    ```

- **Modernization-Based Substitution Rules**
  - **Rule:** Equipment can only be substituted by equal or more modern equipment
  - **Assumption:** Substituting newer equipment for older provides superior or equal capability
  - **Business Logic:** Prevent capability degradation through substitution

- ***SAS:***

```
%if %eval(&modern_subs. > 0) %then %do;
    /* Restrict substitution to equal/higher modernization levels */
    for {<d,c,u,l,s> in ERC_P_SUBASSIGN_SET}
        if mod_level[s]< mod_level[l] then fix erc_p_subassign[ d, c, u, l, s] = 0;
%end;
```

- **Substitution Penalty Structure**
  - **Rule:** Slight preference for primary equipment over substitutes
  - **Assumption:** Primary equipment provides optimal performance for intended role
  - **Implementation:** Penalty effectively negligible in most optimization scenarios
  - ***SAS:***

```
%let sub_assign_pen = .01; /* Minimal substitution penalty */
```

5. DARPL (Priority) Business Rules

- **Unit Priority Integration**
  - **Rules:**
    - Unit priority (DARPL) directly multiplies shortage penalty impact
    - Higher DARPL units receive preferential equipment allocation
  - **DARPL Scale:** 1-100,000 (higher numbers = higher priority)
  - **Assumptions:**
    - DARPL accurately reflects operational importance across mission types
    - Priority differences justify significant resource allocation disparities
  - ***SAS:***

```
/* DARPL priority multiplies shortage penalties */
sum {(c,u,l,e,d) in ERC_P_ASSIGN_SET} (p_pri * DARPL[c,u] * short_ps[c,u,l,d])
```

- **Priority Impact Example:**
  - High Priority Unit (DARPL $= 95{,}000$): $10{,}000{,}000{,}000 \times 95{,}000 = 9.5 \times 10^{14}$ penalty per shortage
  - Low Priority Unit (DARPL $= 5{,}000$): $10{,}000{,}000{,}000 \times 5{,}000 = 5.0 \times 10^{13}$ penalty per shortage

6. Inventory Management Business Rules

- **Inventory Conservation Constraints**
  - **Rules:**
    - Total allocations cannot exceed available inventory plus inbound transfers
    - All inventory must be either assigned, transferred, or held as excess
    - No inventory can be created or destroyed (conservation of mass)
  - **Assumption:** Inventory data accuracy and completeness across all sources
  - ***SAS:***

```
Con INV_MGMT_Y1A {d in dates inter {&min_year.}, c in compos, l in lins}:
    sum{assignments} + sum{transfers_out} + excess[d,c,l] =
        inv_avail[d,c,l] + add_inv[d,c,l];
```

- **Procurement Integration Rules**
  - **Rules:**
    - Procurements cumulate over time and become available inventory
    - New procurements can be allocated in year of delivery or later
  - **Assumptions:**
    - Procurement delivery schedules are accurate and reliable
    - Delivered equipment is immediately available for allocation
  - ***SAS:***

```
for {d in dates, c in compos, l in lins}
    lin_procs[d,c,l] = (sum{v in dates inter {&min_year...d}}(procs[v,c,l]));
```

7. **Data Quality and Processing Business Rules**
   - **LIN Filtering Rules**
     - **Rules:**
       - LINs included only if they have requirements, inventory, procurements, or transfers
       - Zero-quantity records across all categories are excluded from optimization
     - **Assumption:** Excluded LINs represent data errors or inactive equipment
     - ***SAS:***

```
/* LIN inclusion criteria */
if reqd + procs + inventory + xfer_in + xfer_out > 0;
```

   - **Component Standardization Rules**
     - **Rules:**
       - Multiple component identifiers standardized to numeric codes
       - Records with unrecognized component codes are excluded
     - **Assumption:** Component mapping captures all valid Army organizational structures
     - ***SAS:***

```
if compo in ('1', '01', 'AC') then compo_std = '1';
else if compo in ('2', '02', 'ARNG', 'NG') then compo_std = '2';
else if compo in ('3', '03', 'USAR', 'AR') then compo_std = '3';
else if compo in ('6', '06', 'APS') then compo_std = '6';
else delete;
```

   - **Modernization Level Processing**
     - **Rule:** Only equipment with Modernization Level 3 or higher processed in analysis
     - **Business Logic:** Focus on equipment worthy of retention and investment
     - **Assumption:** ML1-2 equipment represents candidates for divestiture
     - ***SAS:***

```
/*** Exclude LINs with mod levels less than 3 per policy ***/
where b.mod_level > 2;
```

8. **Clustering and Optimization Business Rules:**

See 'Clustering Algorithm (Graph-Based Connected Components)' for more detailed description of the clustering process.

   - **Equipment Clustering Rules**
     - **Rules:**
       - Equipment clustered by substitution relationships into connected components
       - Clusters must contain at least one LIN with requirements to be processed
       - Self-substitution relationships excluded from clustering
     - **Assumption:** Substitutable equipment can be optimally allocated together
     - ***SAS:***

```
/* Connected components clustering logic */
if (c_lin ne "" and s_lin ne "") and source ne 7 and c_lin ne s_lin;
```

   - **Constraint Tightening Rules**
     - **Rules:**
       - Decision variables fixed to zero when underlying data is zero
       - Optimization space reduced through constraint tightening
     - **Assumption:** Zero requirements cannot generate shortages or assignments
     - ***SAS:***

```
/* Fix decision variables for zero requirements */
for {<d,c,u,l> in ERC_P_ASSIGN_SET}
    if reqd_P[d,c,u,l] = 0 then fix Short_Ps[d,c,u,l] = 0;
```

9. **Temporal Business Rules**
   - **Multi-Year Optimization Horizon**
     - **Rules:**
       - Optimization considers 7-year planning horizon
       - Decisions in early years affect inventory availability in later years

- **Assumption:** Requirements and procurement projections are reliable across planning horizon
- ***SAS:***

```
set <number> modeling_dates = {&first_year...&last_year.};
```

- **First Year Constraint Rules**
    - **Rule:** Option to freeze first 1-2 years of optimization to current state
    - **Business** Logic: Recognize practical limitations of immediate equipment redistribution
    - **Assumption:** Current allocations represent operationally acceptable baseline
    - ***SAS:***

```
%if %eval(&freeze_first_two. = 1) %then %do;
    /* Fix first two years to current allocations */
    Con Freeze_Year_1{...}: assignments = current_inventory;
%end;
```

10. Reporting and Audit Business Rules

- **Congressional Reporting Requirements**
    - **Rules:**
        - Complete audit trail maintained for all optimization runs
        - All parameter values documented for Congressional transparency
        - Solution reproducibility required for decision justification
    - **Assumption:** Detailed documentation supports policy and budget decisions
    - ***SAS:***

```
/* Comprehensive audit trail generation */
data run_parameters;
    execution_timestamp = datetime();
    /* Document all optimization parameters for reproducibility */
run;
```

- **Output Validation Rules**
    - **Rules:**
        - Zero-value results excluded from management reports
        - Focus reporting on actionable inventory imbalances
    - **Assumption:** Zero values represent noise rather than meaningful information
    - ***SAS:***

```
/* Filter positive results for reporting */
where Short_Ps > 0;  /* Only report actual shortages */
where excess > 0;    /* Only report actual excess */
```

11. System Performance and Reliability Assumptions

- **Computational Assumptions**
    - Optimization problems are computationally tractable within time constraints
    - Integer linear programming provides globally optimal solutions
    - Solution quality remains high as problem size scales
- **Data Integration Assumptions**
    - Multiple data sources (SACS, LDAC, LMDB) provide consistent and compatible information
    - Data refresh cycles align with decision-making timelines
    - Source system reliability supports mission-critical analysis
- **Business Process Assumptions**
    - Optimization recommendations can be implemented through existing Army processes
    - Equipment transfers can be executed within modeled timeframes
    - Unit commanders will accept optimally allocated equipment

Clustering Algorithm (Graph-Based Connected Components):

Description: The clustering algorithm groups equipment items (LINs) that can substitute for each other into connected clusters. Think of it like finding groups of friends where if Person A knows Person B, and Person B knows Person C, then A, B, and C are all in the same social network cluster - even if A and C don't directly know each other.

For equipment: if Tank A can substitute for Tank B, and Tank B can substitute for Tank C, then A, B, and C form one cluster that can be optimized together, even if Tank A cannot directly substitute for Tank C.

1. **Graph Structure**
   - **Nodes (Vertices):**
     - Each equipment LIN (Line Item Number) is a node
     - Only LINs with inventory, requirements, or procurement data are included
   - **Edges (Links):**
     - Each valid substitution relationship creates an undirected edge
     - Edge exists between LIN A and LIN B if A can substitute for B OR B can substitute for A
   - **Graph Properties:**
     - **Undirected :** Substitution relationships create bidirectional connections
     - **Unweighted :** All substitution relationships have equal graph weight
     - **Simple :** No self-loops or multiple edges between same nodes

2. **Pseudo-Code Algorithm**

ALGORITHM: Equipment_Clustering_Connected_Components

- **INPUT:**
  - relevant_lins: Set of LINs with inventory/requirements/procurement data
  - substitution_rules: List of (source_lin, substitute_lin) pairs
- **OUTPUT:**
  - clusters: Mapping of each LIN to its cluster ID

STEP 1: Build Graph Structure
- nodes = relevant_lins
- edges = empty_set

- FOR each substitution_rule in substitution_rules:
  - source = substitution_rule.lins
  - substitute = substitution_rule.sublins

  - IF source IN relevant_lins AND substitute IN relevant_lins:
    - IF source != substitute: // Avoid self-loops
      - edges.add((source, substitute))

STEP 2: Find Connected Components
- visited = empty_set
- cluster_id = 0
- cluster_mapping = empty_map

- FOR each node in nodes:
    - IF node NOT IN visited:
        - cluster_id = cluster_id + 1
        - component_nodes = depth_first_search(node, edges, visited)

    - FOR each component_node in component_nodes:
        - cluster_mapping[component_node] = cluster_id

STEP 3: Filter Valid Clusters

- valid_clusters = empty_set

- FOR each cluster_id in unique_cluster_ids:
    - cluster_lins = all_lins_with_cluster_id(cluster_id)
    - IF any_lin_has_requirements(cluster_lins):
        - valid_clusters.add(cluster_id)

- RETURN clusters WHERE cluster_id IN valid_clusters

- FUNCTION depth_first_search(start_node, edges, visited):
    - stack = [start_node]
    - component = empty_set

    - WHILE stack NOT empty:
        - current = stack.pop()

    - IF current NOT IN visited:
        - visited.add(current)
        - component.add(current)

    - neighbors = all_nodes_connected_to(current, edges)
    - FOR each neighbor in neighbors:
        - IF neighbor NOT IN visited:
            - stack.append(neighbor)

- RETURN component

3. Mathematical Formulation

- **Graph Definition:**
    - $G = (V, E)$ where:
        - $V = \{lin \in LINs \mid \exists(inventory(lin) > 0) \vee (requirements(lin) > 0) \vee (procurement(lin) > 0)\}$
        - $E = \{(u, v) \mid \exists \text{ substitution\_rule}(u, v) \vee \text{substitution\_rule}(v, u), u \neq v, u, v \in V\}$
- **Connected Components:**
    - $C = \{C_1, C_2, \ldots, C_k\}$ where:
        - $C_i \subseteq V$ for all i
        - $C_i \cap C_j = \emptyset$ for $i \neq j$ (disjoint)
        - $\bigcup_i C_i = V$ (complete coverage)
        - $\forall u, v \in C_i, \exists$ path $P = (u = v_0, v_1, \ldots, v_n = v)$ where $(v_j, v_{j+1}) \in E$
- **Path Connectivity:**
    - $\text{connected}(u, v) \equiv \exists$ path $P = (u = v_0, v_1, v_2, \ldots, v_n = v)$
        - such that $\forall j \in \{0, 1, \ldots, n-1\}: (v_j, v_{j+1}) \in E$

- **Cluster Validation:**

```
/* Connected components analysis using SAS/OR */
proc optgraph
    data_nodes = relevant_lins
    data_links = substitution_network;

    connected_components
        nodes_out = connected_nodes(partition=component_id)
        summary_out = component_summary;
run;
```

- **Implementation in R (Target):**

```
# Graph-based clustering using igraph
cluster_equipment_substitutions <- function(substitution_rules, relevant_lins) {

  library(igraph)

  # Create undirected graph
  substitution_graph <- graph_from_data_frame(
    d = substitution_rules[, c("lins", "sublins")],
    vertices = relevant_lins,
    directed = FALSE
  )

  # Find connected components
  components <- components(substitution_graph)

  # Return cluster mapping
  return(data.frame(
    lins = V(substitution_graph)$name,
    cluster_id = components$membership
  ))
}
```

Penalty Structure Details:

This penalty structure ensures that the optimization model prioritizes:

- Filling ERC P requirements for high-DARPL units first
- Minimizing inter-component transfers
- Deploying newer equipment rather than storing as excess
- Using primary equipment over substitutes when possible

- **Primary Penalty Categories**
  - **Shortage Penalties (ERC-Based)**
    - **ERC P (Primary) Shortages:**
      - **Purpose:** Heavily penalize shortages of mission-critical primary equipment
      - **Application:** Applied to all ERC P requirement shortfalls
      - **Formula:** $\mathbf{DARPL[c,u] \times p\_pri \times shortage[c,u,l,'P',d]}$
      - **SAS:**

```
%let p_pri = 10000000000;  /* 10 billion penalty multiplier */
```

    - **ERC A (Augmentation) Shortages:**
      - **Purpose:** Moderate penalty for secondary equipment shortages
      - **Application:** Applied to all ERC A requirement shortfalls
      - **Formula:** $\mathrm{DARPL[c,u]} * \mathrm{a\_pri} * \mathrm{shortage[c,u,l,'A',d]}$
      - ***SAS:***

```
%let a_pri = 500;          /* 500 penalty multiplier */
```

    - **ERC B/C (Background/Commercial) Shortages:**
      - **Purpose:** Minimal penalties for non-critical equipment categories
      - **Note:** Generally filtered out during data cleaning process
      - ***SAS:***

```
%let b_pri = 100;          /* 100 penalty multiplier */
%let c_pri = 10;           /* 10 penalty multiplier */
```

  - **Transfer Penalties (Inter-Component)**
    - **Component Transfer Penalty:**
      - **Purpose:** Discourage unnecessary equipment movement between components
      - **Application:** Applied to all inter-component transfers
      - **Components:** $\mathbf{AC\ (1) \leftrightarrow ARNG\ (2) \leftrightarrow USAR\ (3) \leftrightarrow APS\ (6)}$
      - **Formula:** $\mathbf{trans\_pen \times transfer[c1,c2,l,e,d]}$
      - **SAS:**

```
%let trans_pen = 150;      /* Transfer penalty per unit */
```

  - **Modernization Level Excess Penalties**
    - **Modernization Level 5 (Newest) Excess:**
      - **Purpose:** Heavily discourage holding newest equipment as excess
      - **Rationale:** Newest equipment should be deployed, not stored
      - ***SAS:***

```
%let mod_5_e_pen = 999;    /* Highest excess penalty */
```

    - **Modernization Level 4 Excess:**
      - **Purpose:** Discourage excess of near-newest equipment
      - ***SAS:***

```
%let mod_4_e_pen = 499;    /* High excess penalty */
```

- **Modernization Level 3 Excess:**
  - **Purpose:** Moderate discouragement of mid-level equipment excess
  - ***SAS:***

```
%let mod_3_e_pen = 200;     /* Moderate excess penalty */
```

- **Modernization Level 2 Excess:**
  - **Purpose:** Minimal penalty for older equipment excess
  - ***SAS:***

```
%let mod_2_e_pen = 10;     /* Low excess penalty */
```

- **Modernization Level 1 (Oldest) Excess:**
  - **Purpose:** Very low penalty for oldest equipment excess
  - ***SAS:***

```
%let mod_1_e_pen = 5;     /* Minimal excess penalty */
```

- **Substitution Assignment Penalty**
  - **Substitution Usage Penalty:**
    - **Purpose:** Slight preference for primary equipment over substitutes
    - **Application:** Applied when using substitution rules
    - **Note:** Effectively inactive due to very small value
    - **SAS:**

```
%let sub_assign_pen = .01; /* Very small substitution penalty */
```

- **Unit Inventory Transfer Penalty**
  - **Yearly Transfer Penalty:**
    - **Purpose:** Penalize frequent equipment transfers between units
    - **Application:** Applied to unit-level inventory movements
    - **Configuration:** Set via run parameters
    - ***SAS:***

```
%let unit_yearly_xfer_pen = [configured per run];
```

- **Penalty Hierarchy and Optimization Logic**
  - **Priority Ranking (Highest to Lowest Impact)**
    - ERC P Shortages: 10,000,000,000 × DARPL priority
    - ERC A Shortages: 500 X DARPL priority
    - Modernization Level 5 Excess: 999 per unit
    - Modernization Level 4 Excess: 499 per unit
    - Modernization Level 3 Excess: 200 per unit
    - Inter-Component Transfers: 150 per unit
    - ERC B Shortages: 100 × DARPL priority
    - ERC C Shortages: 10 × DARPL priority
    - Modernization Level 2 Excess: 10 per unit
    - Modernization Level 1 Excess: 5 per unit
    - Substitution Usage: 0.01 per substitution
    - Unit Yearly Transfers: Variable per configuration

- **Mathematical Implementation in Objective Function**
- ***SAS:***

```
/* Complete objective function with all penalty terms */
con Obj:
    /* Primary shortage penalties */
    sum {(c,u,l,e,d) in ERC_P_ASSIGN_SET}
        (p_pri * short_ps[c,u,l,d]) +
```

```
/* Augmentation shortage penalties */
sum {(c,u,l,e,d) in ERC_A_ASSIGN_SET}
    (a_pri * short_as[c,u,l,d]) +

/* Transfer penalties */
sum {(c1,c2,u,l,d) in TRANSFER_SET}
    (trans_pen * transfer[c1,c2,u,l,d]) +

/* Excess inventory penalties by modernization level */
sum {(c,l,d) in INVENTORY_SET}
    (mod_5_e_pen * excess_mod5[c,l,d] +
     mod_4_e_pen * excess_mod4[c,l,d] +
     mod_3_e_pen * excess_mod3[c,l,d] +
     mod_2_e_pen * excess_mod2[c,l,d] +
     mod_1_e_pen * excess_mod1[c,l,d]) +

/* Substitution penalties */
sum {(c,u,l,s,d) in SUBSTITUTION_SET}
    (sub_assign_pen * substitution[c,u,l,s,d]);
```

- **DARPL Priority Integration**
  The Departmental Army Priority List (DARPL) provides unit-specific priority multipliers that amplify shortage penalties:
    - **DARPL Formula Application:**
        - **Final Penalty** = Base Penalty X DARPL Priority X Shortage Quantity
        - **DARPL Range:** 1-100,000 (higher numbers = higher priority)
        - **Effect:** Creates dramatic penalty differences between high and low priority units
    - **Example Calculation:**
        - High Priority Unit (DARPL = 95,000): 10,000,000,000 X 95,000 = 9.5 X 10^{14} penalty per ERC P shortage
        - Low Priority Unit (DARPL = 5,000): 10,000,000,000 X 5,000 = 5.0 X 10^{13} penalty per ERC P shortage