```
% Initialize the system
clear all
close all
clc

resources
machines
processes
```

```
% Construct our problem

% Set the desired output rate
b = new_rate_vector({"electronic_circuit", 100});
disp("Model Desired Output:");
```

```
Model Desired Output:
```

```
display_rates(b);
```

```
Non-zero rates:
electronic_circuit: 100
```

```

% Limit the resource scope
resource_scope = ["copper_ore" "iron_ore" "copper_plate" "iron_plate"
"copper_cable" "electronic_circuit"]';
[resource_indices,~] = item_names_to_indices(resource_scope);
% Limit the process scope
[process_indices,~] = process_names_to_indices(["copper_mining" "iron_mining"
"copper_plate_smelting" "iron_plate_smelting" "copper_cable_crafting"
"circuit_crafting"]');

% Grab the subsets
A_ = A(resource_indices,process_indices);
b_ = b(resource_indices);

% Solve the reduced system
x_ = A_\b_;

% x_ is the solution to the reduced system, so we have to expand it back up
% to the full space.
x = zeros(size(A,2),1);
for i=1:length(process_indices)
    x(process_indices(i)) = x_(i);
end

% With system solved based on the constraints, we can take that solution to
% the sub-system and compute it with the whole system. This is important to
% check the result.
```

```
a=A*x;

x2 = assign_default_machines_to_unit_processes(x);

disp("Model Output:");
```

```
Model Output:
```

```
display_rates(a);
```

```
Non-zero rates:
electronic_circuit: 100
```

```
display_processes(x);
```

```
Non-zero UNIT processes:
iron_mining    100
copper_mining    150
iron_plate_smelting    320
copper_plate_smelting    480
copper_cable_crafting    75
circuit_crafting    50
```

```
display_processes(x2);
```

```
Non-zero processes:
iron_mining    electric_mining_drill    200
copper_mining    electric_mining_drill    300
iron_plate_smelting    electric_furnace    160
copper_plate_smelting    electric_furnace    240
copper_cable_crafting    assembling_machine_3    60
circuit_crafting    assembling_machine_3    40
```