```
int N = 5;
int M = 4;

range Products = 1..N;
range Machines = 1..M;

int profit[Products] = [40, 70, 20, 80, 140];
int resources_usage[Products] = [2, 3, 2, 4, 6];
int total_resources = 20;
int machine_rental_cost[Machines] = [50, 60, 15, 45];
int machines_products_matrix[Machines][Products] = [[0,1,1,1,0],[1,1,0,1,1],[0,0,1,0,0],[1,0,0,0,1]];

dvar int+ x[Products];
dvar boolean y[Machines];

maximize
 sum(i in Products,j in Machines) (profit[i] * x[i] - machine_rental_cost[j] * y[j]);

subject to {


   forall(i in Products) ((x[i] != 0) <= sum(j in Machines)(machines_products_matrix[j][i]*y[j]));
    sum(i in Products) resources_usage[i] * x[i] <= total_resources;
    sum(j in Machines) y[j] >= 1;
    sum(i in Products) x[i] >= 1;




}

execute DISPLAY {
    writeln("Optymalne rozwiązanie:");
    write("Wynajmujemy maszyny nr ");
    for (j in Machines){
      if (y[j] == 1) write(j," ");
    }

    writeln("by produkować");
    for (i in Products) {
        writeln( x[i], " jednostek produktu ", i);
    }
}
```