

100

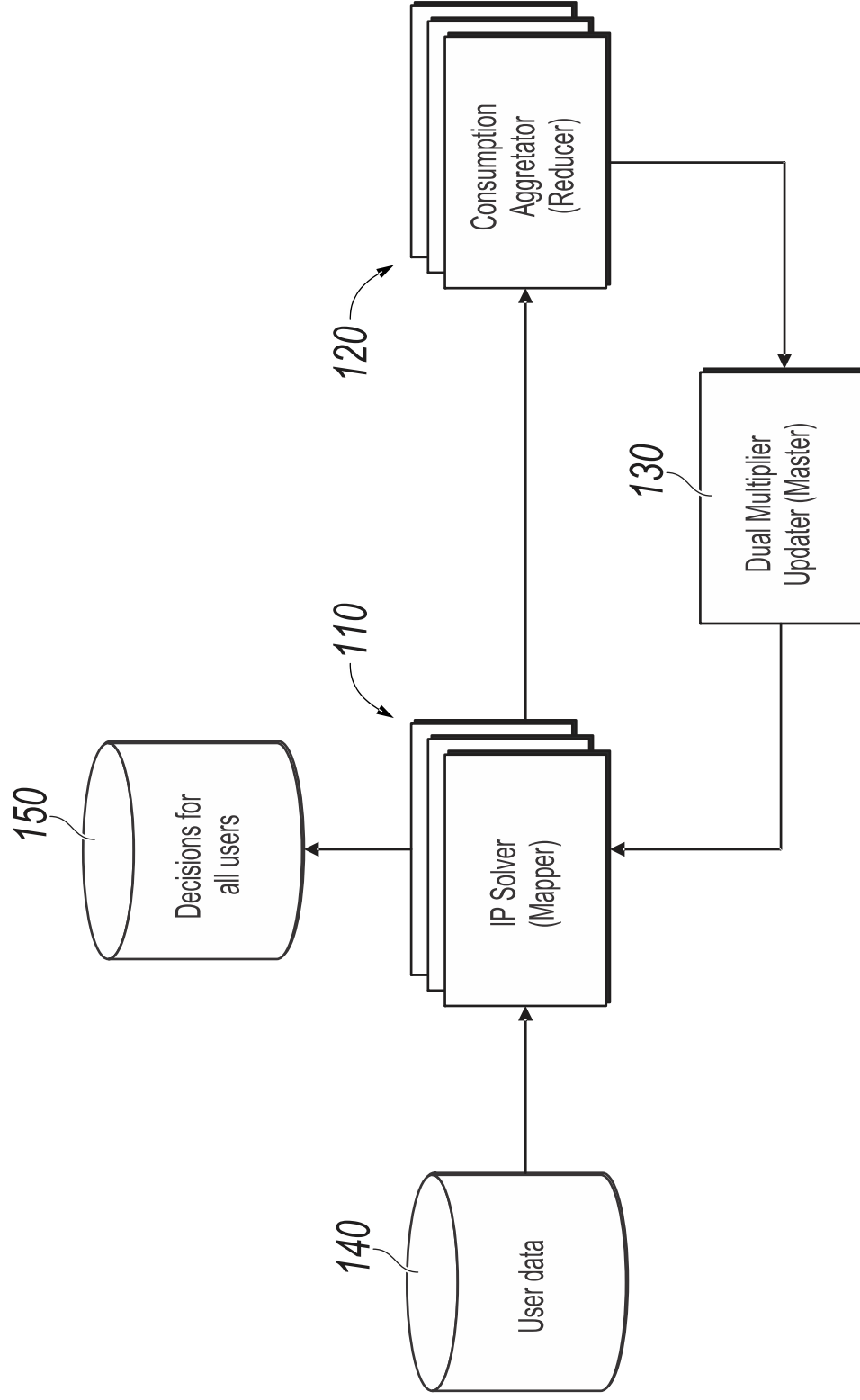


FIG. 1



Algorithm 1

210 { **Procedure** Map (p_i, b_i, c_i, λ^t)
 | solve $x_{i,j}$ for (11)
 | **for each** $k \in \{1, \dots, K\}$ **do**
 | | $\text{value} = \sum_{j=1}^M b_{i,j,k} x_{i,j}$
 | | **emit**(k, value)
 | **end**

220 { **Procedure** Reduce ($k, [\text{value}]$)
 | **return** Σvalue

250 { initialize λ^0
 | **for** $t \leftarrow 0$ **to** $T - 1$ **do**
 | | # Parallel
 | | **for each** $i \in \{1, \dots, N\}$ **do** ← 252
 | | | Map (p_i, b_i, c_i, λ^t)
 | | **end**
 | | # Parallel
 | | **for each** $k \in \{1, \dots, K\}$ **do** ← 254
 | | | $R_k = \text{Reduced}(k, [\text{value}])$
 | | **end**
 | | **for each** $k \in \{1, \dots, K\}$ **do** ← 256
 | | | $\lambda_k^{t+1} = \lambda_k^t + \alpha(R_k - B_k)$
 | | | $\lambda_k^{t+1} = \max(0, \lambda_k^{t+1})$
 | | **end**
 | | **if** λ^t has converged **then** ← 258
 | | | **return** λ^t
 | | **end**
 | **end**
 | **return** λ^T

FIG. 2

300

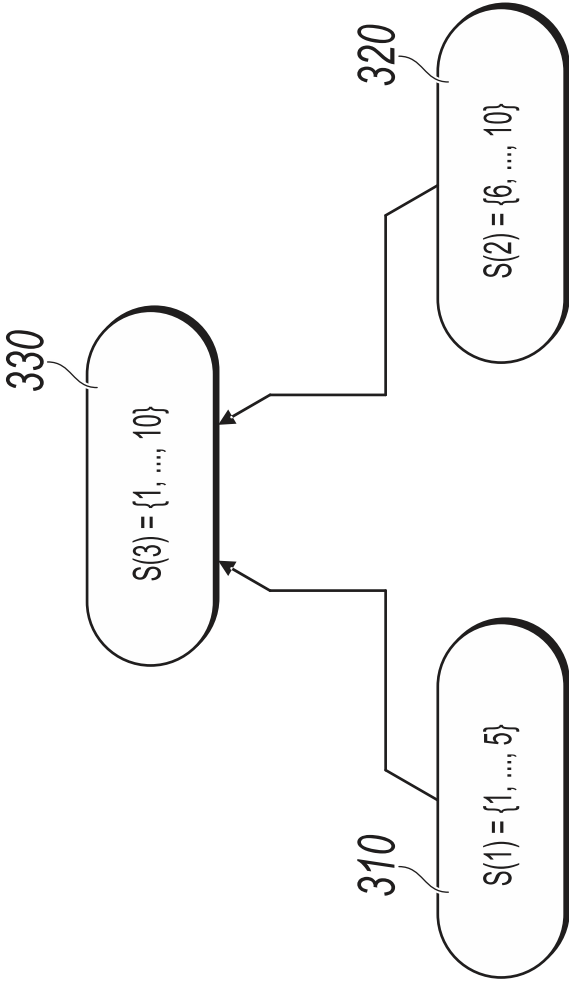


FIG. 3



Algorithm 2

```

410 → initialize  $x_{i,j}$  for all  $j = 1, \dots, M$ 
      sort indices  $j \in \{1, \dots, M\}$  in a non-increasing order of  $\tilde{p}_{i,j}$  ← 420
430 → for  $S(l)$  in a topological ordering of the DAG do
      |   fetch the indices  $j \in S(l)$  with  $x_{i,j} = 1$  ← 440
450 → |   update  $x_{i,j} = 0$  if  $j$  is not in top  $C_l$  sorted indices
      end
      return  $x_{i,j} \ j = 1, \dots, M$  ← 460
  
```

FIG. 4

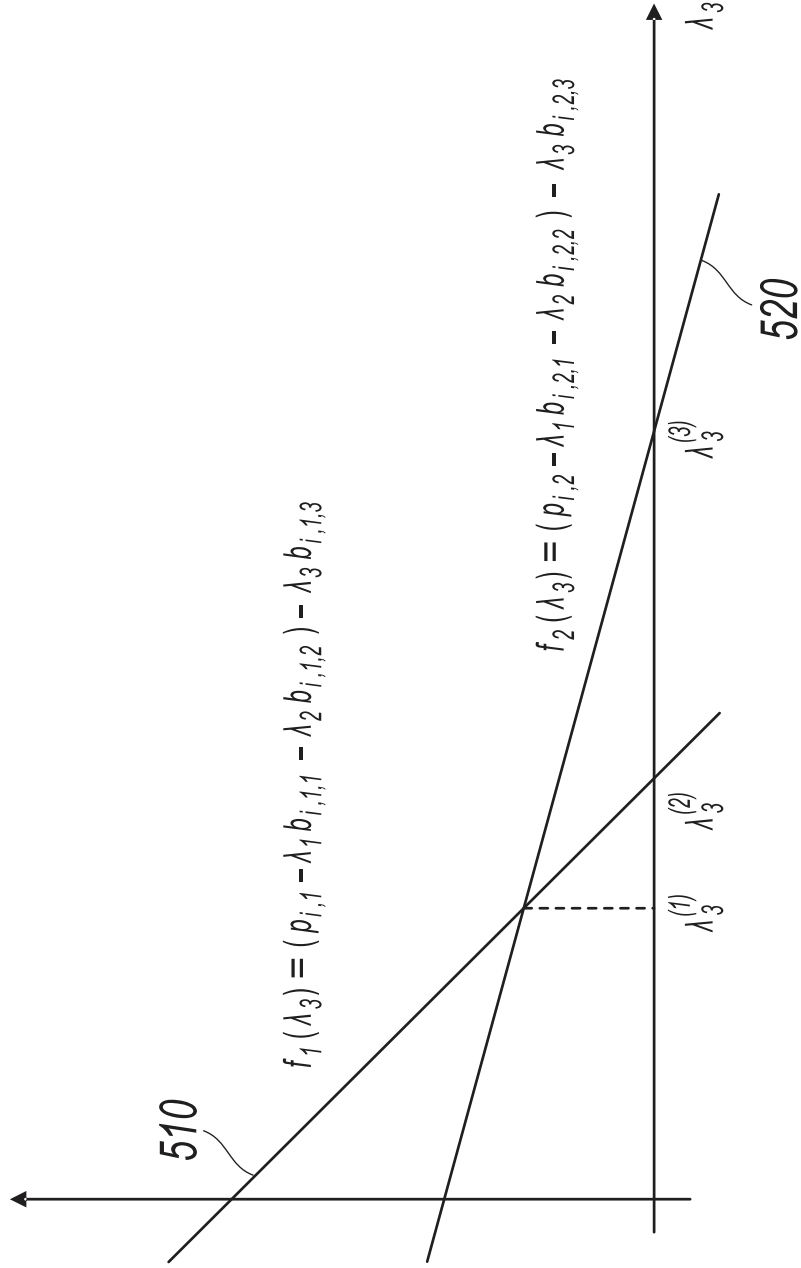


FIG. 5



Algorithm 3

```

# Procedure Intersect (  $f, g$ ) returns a non-negative  $\lambda_k$ 
# at which  $f(\lambda_k) = g(\lambda_k) \geq 0$  when  $f$  and  $g$  are not parallel
Procedure CalculateIntersectionPoints (  $p_i, b_i, \lambda^t, k$ )
     $\Lambda_k = \emptyset$ 
    for each  $j \in \{1, \dots, M\}$  do
         $f_j(\lambda_k) = (p_{i,j} - \sum_{k'=1, k' \neq k}^K \lambda_{k'}^t b_{i,j,k'}) - \lambda_k b_{i,j,k}$ 
         $g(\lambda_k) = 0$  # This is the horizontal axis.
         $\Lambda_k = \Lambda_k \cup \text{Intersect}(f_j, g)$ 
        for each  $j' \in \{j+1, \dots, M\}$  do
             $f_{j'}(\lambda_k) = (p_{i,j'} - \sum_{k'=1, k' \neq k}^K \lambda_{k'}^t b_{i,j',k'}) - \lambda_k b_{i,j',k}$ 
             $\Lambda_k = \Lambda_k \cup \text{Intersect}(f_j, f_{j'})$ 
        end
    end
return the set of non-duplicate values in  $\Lambda_k$ 

```

FIG. 6

710 {

```

Procedure Map ( $p_i, b_i, c_i, \lambda^t$ )
  for each  $k \in \{1, \dots, K\}$  do
     $\Lambda_k = \text{CalculateIntersectionPoints}(p_i, b_i, \lambda^t, k)$  ← 712
    sort  $\Lambda_k$  in a decreasing order ← 714
    previous_sum = 0
    for each  $\lambda_k$  in  $\Lambda_k$  do
       $\lambda = (\lambda_1^t, \dots, \lambda_{k-1}^t, \lambda_k, \lambda_{k+1}^t, \dots, \lambda_K^t)$ 
      solve  $x_{i,j}$  with multipliers  $\lambda$  using Algorithm 2 ← 715
      current_sum =  $\sum_{j=1}^M b_{i,j,k} x_{i,j}$  ← 716
      if current_sum == previous_sum then
        | continue
      end
       $v_1 = \lambda_k$ 
       $v_2 = \text{current\_sum} - \text{previous\_sum}$ 
      value =  $[v_1, v_2]$ 
      emit( $k, \text{value}$ ) ← 718
      previous_sum = current_sum
    end
  end

```

720 {

```

Procedure Reduce( $k, [v_1, v_2]$ )
  if  $\sum v_2 \leq B_k$  then
    | return 0
  else
    | return minimal threshold  $v$  such that  $\sum_{v_1 \geq v} v_2 \leq B_k$  ← 722
  end

```

750 {

```

  initialize  $\lambda^0$ 
  for  $t \leftarrow 0$  to  $T-1$  do
    # Parallel
    for each  $i \in \{1, \dots, N\}$  do ← 752
      | Map( $p_i, b_i, c_i, \lambda^t$ )
    end
    # Parallel
    for each  $k \in \{1, \dots, K\}$  do ← 754
      |  $\lambda_k^{t+1} = \text{Reduce}(k, [v_1, v_2])$ 
    end
    if  $\lambda^t$  has converged then ← 756
      | return  $\lambda^t$ 
    end
  end
  return  $\lambda^T$ 

```

FIG. 7



Algorithm 5

```

Procedure Map ( $p_i, b_i, c_i, \lambda^t$ )
  initialize adjusted_profits as an array of  $K$  numbers
  foreach  $k \in \{1, \dots, K\}$  do
    | adjusted_profits [ $k$ ] =  $\max(p_i, k - \lambda_k^t b_{i,k,k}, 0)$ 
  end
   $Q\_th\_largest$  = quick_select(adjusted_profits,  $Q$ )
   $Q1\_th\_largest$  = quick_select(adjusted_profits,  $Q + 1$ )
  foreach  $k \in \{1, \dots, K\}$  do
    | if adjusted_profits [ $k$ ]  $\geq Q\_th\_largest$  then
    | |  $\bar{p} = Q1\_th\_largest$ 
    | else
    | |  $\bar{p} = Q\_th\_largest$ 
    | end
    | if  $p_{i,k} > \bar{p}$  then
    | |  $v_1 = \frac{p_{i,k} - \bar{p}}{b_{i,k,k}}$ 
    | |  $v_2 = b_{i,k,k}$ 
    | | value = [ $v_1, v_2$ ]
    | | emit( $k$ , value)
    | end
  end

```

FIG. 8



Algorithm 6

```

 $\lambda^0 = (1.0, \dots, 1.0)$  #  $K$  default values
for  $t \leftarrow 0$  to  $\tau - 1$  do
    sample a small set of  $n$  random groups  $\tilde{G}$ 
     $\tilde{B}_k = \frac{n}{N} B_k$  for all  $k = 1, \dots, K$ 
    if  $t = 0$  then
         $\tilde{\lambda} = \lambda^0$ 
    else
         $\tilde{\lambda} = \frac{1}{t} \sum_{s=1}^t \lambda^s$ 
    end
    compute  $\lambda^{t+1}$  by applying Algorithm 4 to  $\tilde{G}$  with budgets
     $\tilde{B}_k$  and initial multipliers  $\tilde{\lambda}$ 
end
return  $\frac{1}{\tau} \sum_{t=1}^t \lambda^t$ 

```

FIG. 9

1000 ↗

10 / 20

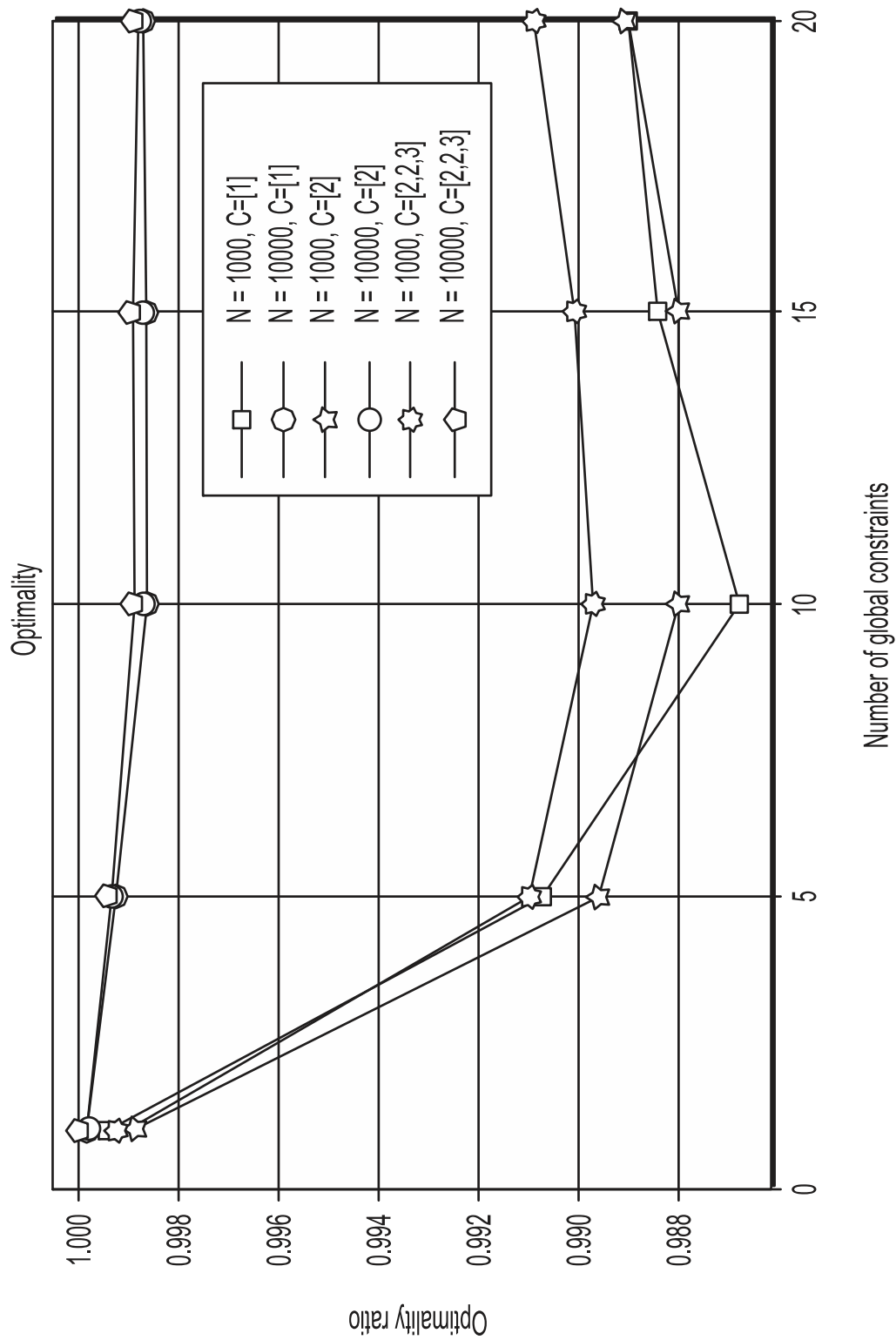


FIG. 10

1100 ↘

Table 1: Number of SCD Iterations with/without Pre-solving

N	no pre-solving	pre-solving	% of reduction
1 million	35	21	40%
10 million	32	8	75%
100 million	32	13	59%

FIG. 11

1200 ↘

Table 2: Results for 100 million users (with up to 10 billion items)

<i>K</i>	1	5	10	20	100
Number of iterations	2	13	18	14	10
Primal value	40,631,183.07	73,000,742.85	85,378,580.47	92,415,786.11	98,436,146.56
Duality gap	0	302.71	290.74	301.04	225.15
Number of constraint violations	0	0	0	0	0

FIG. 12

Assume $N = 3$, $k = 3$, and $B_3 = 28$

Each user solves an IP problem according to Map function 710

1310

User 1:

$\lambda_3^{(1)} = 9, \text{current_sum} = 5$
 $\lambda_3^{(2)} = 6, \text{current_sum} = 6$
 $\lambda_3^{(3)} = 3, \text{current_sum} = 7$

} emit =>

$k, [v_1, v_2]$
 $3, [9, 5]$
 $3, [6, 1]$
 $3, [3, 1]$

User 2:

$\lambda_3^{(1)} = 8, \text{current_sum} = 10$
 $\lambda_3^{(2)} = 5, \text{current_sum} = 12$
 $\lambda_3^{(3)} = 2, \text{current_sum} = 13$

} emit =>

$3, [8, 10]$
 $3, [5, 2]$
 $3, [2, 1]$

User 3:

$\lambda_3^{(1)} = 7, \text{current_sum} = 9$
 $\lambda_3^{(2)} = 4, \text{current_sum} = 10$
 $\lambda_3^{(3)} = 1, \text{current_sum} = 11$

} emit =>

$3, [7, 9]$
 $3, [4, 1]$
 $3, [1, 1]$

Sorting for $k = 3$

Candidates of λ_3

Incremental used resources

1320

9,	5	} sum to 28
8,	10	
7,	9	
6,	1	
5,	2	
4,	1	
3,	1	
2,	1	
1,	1	

$v = 4$

Furthermore, if $B_k = 28.5$, $v = 3.5$ by interpolating

FIG. 13

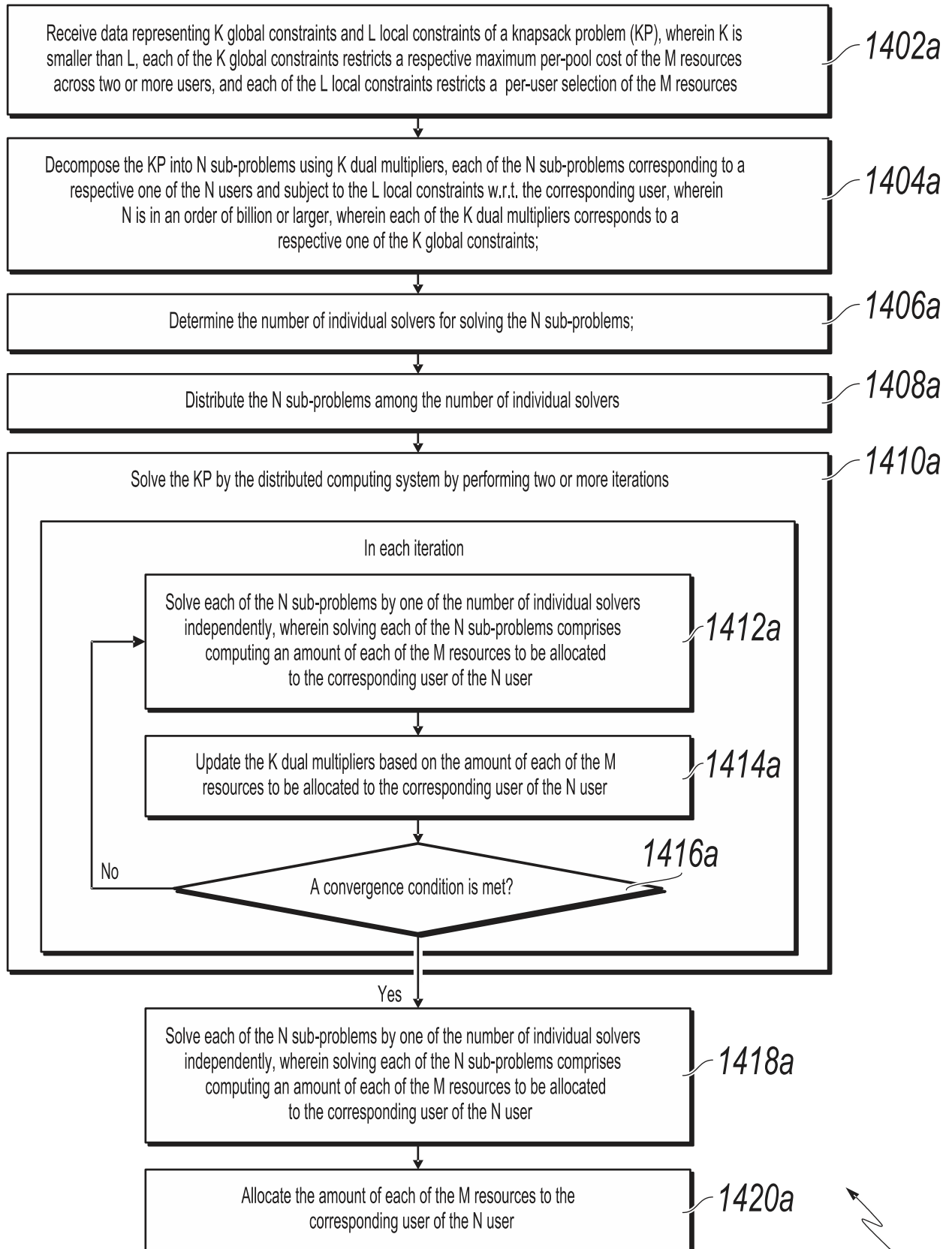


FIG. 14A

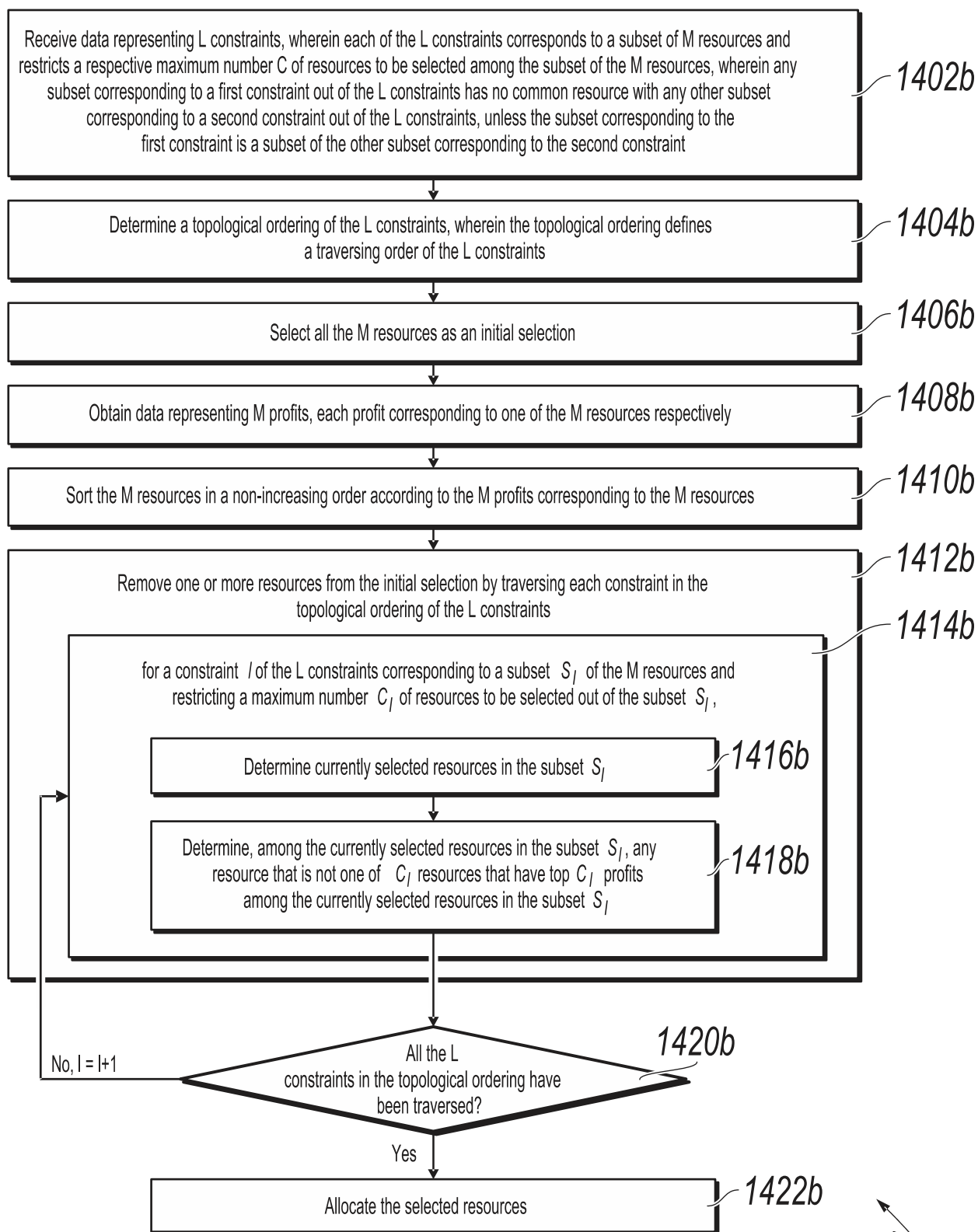


FIG. 14B

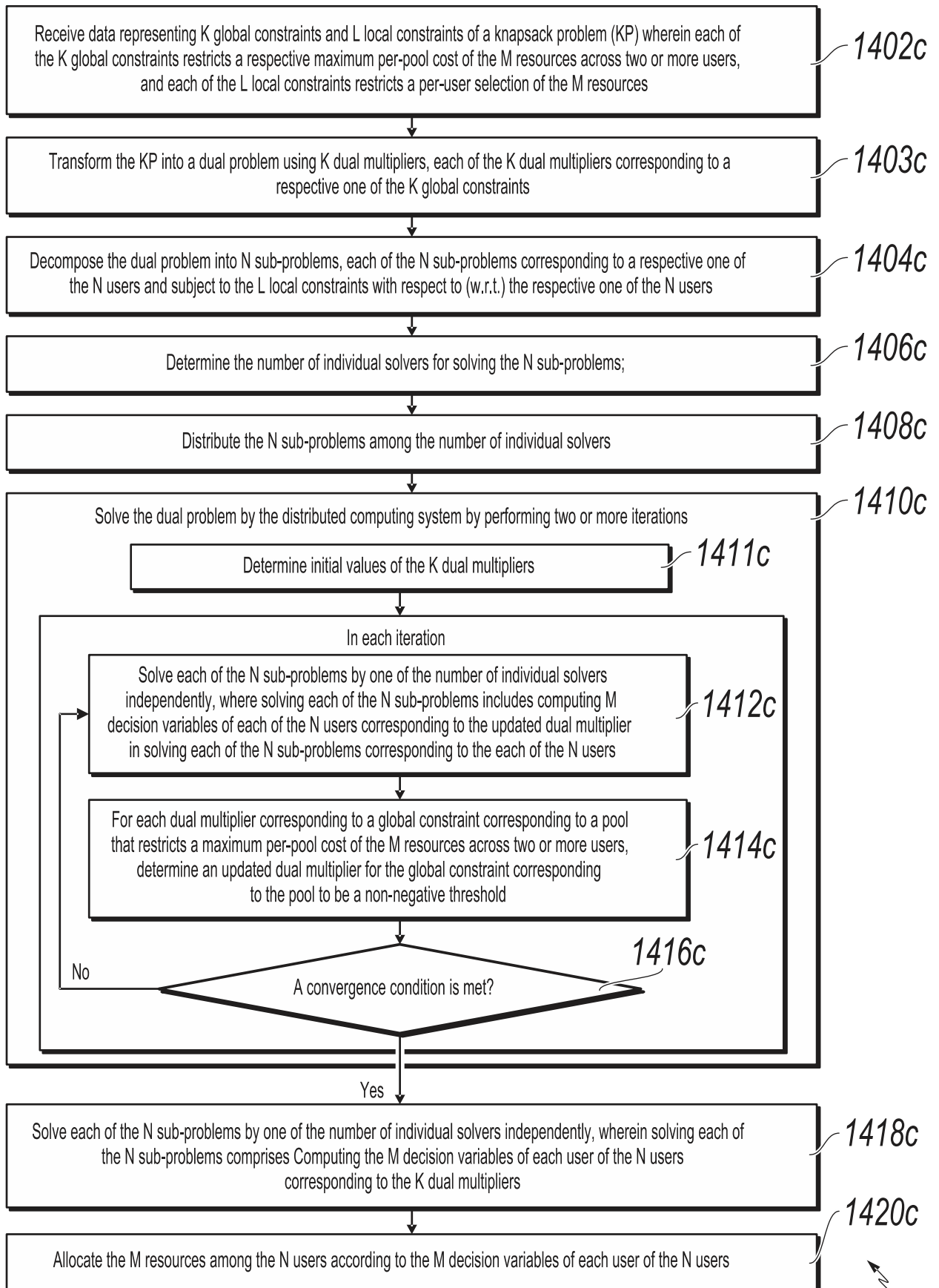


FIG. 14C

1400c

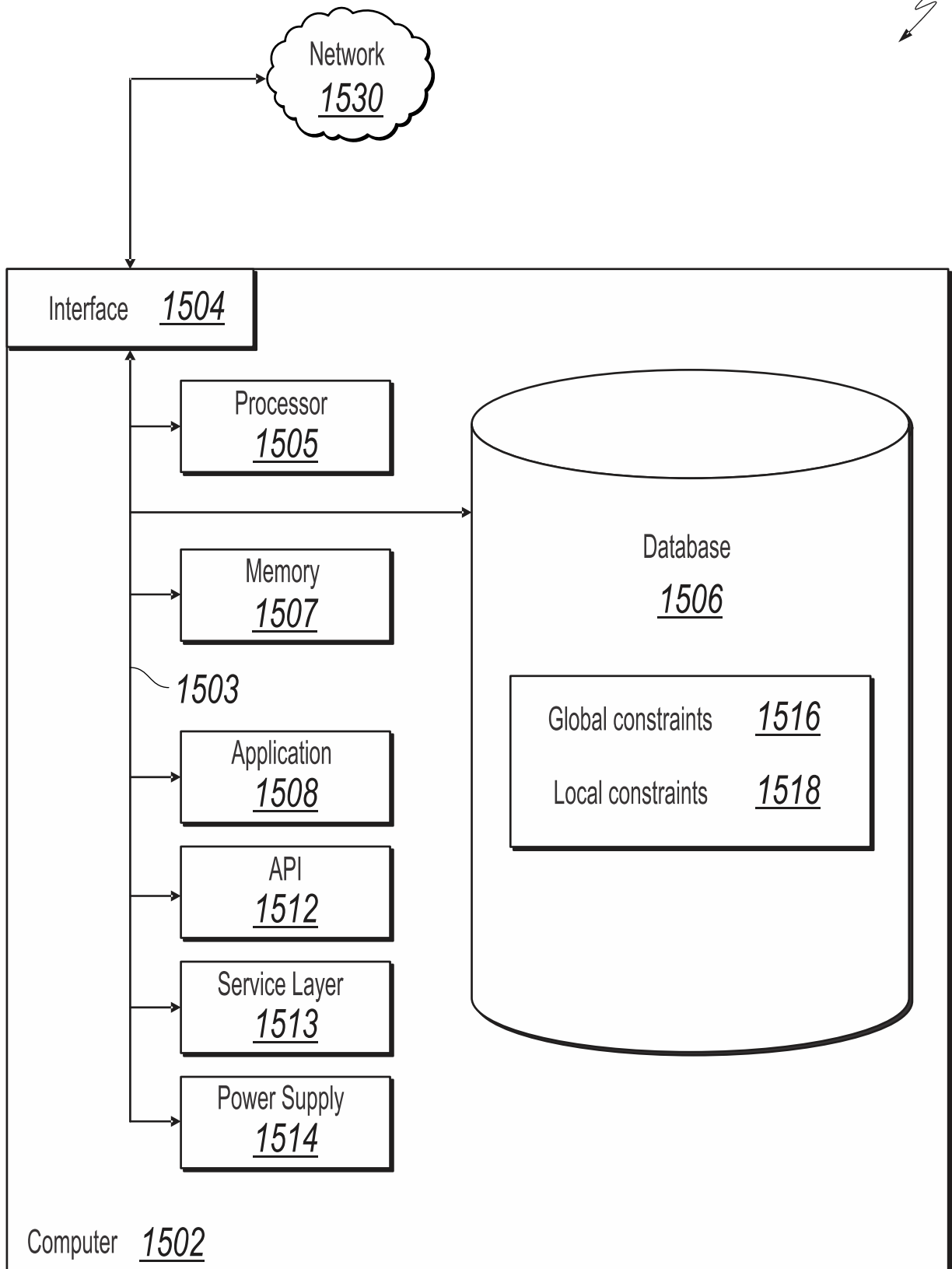
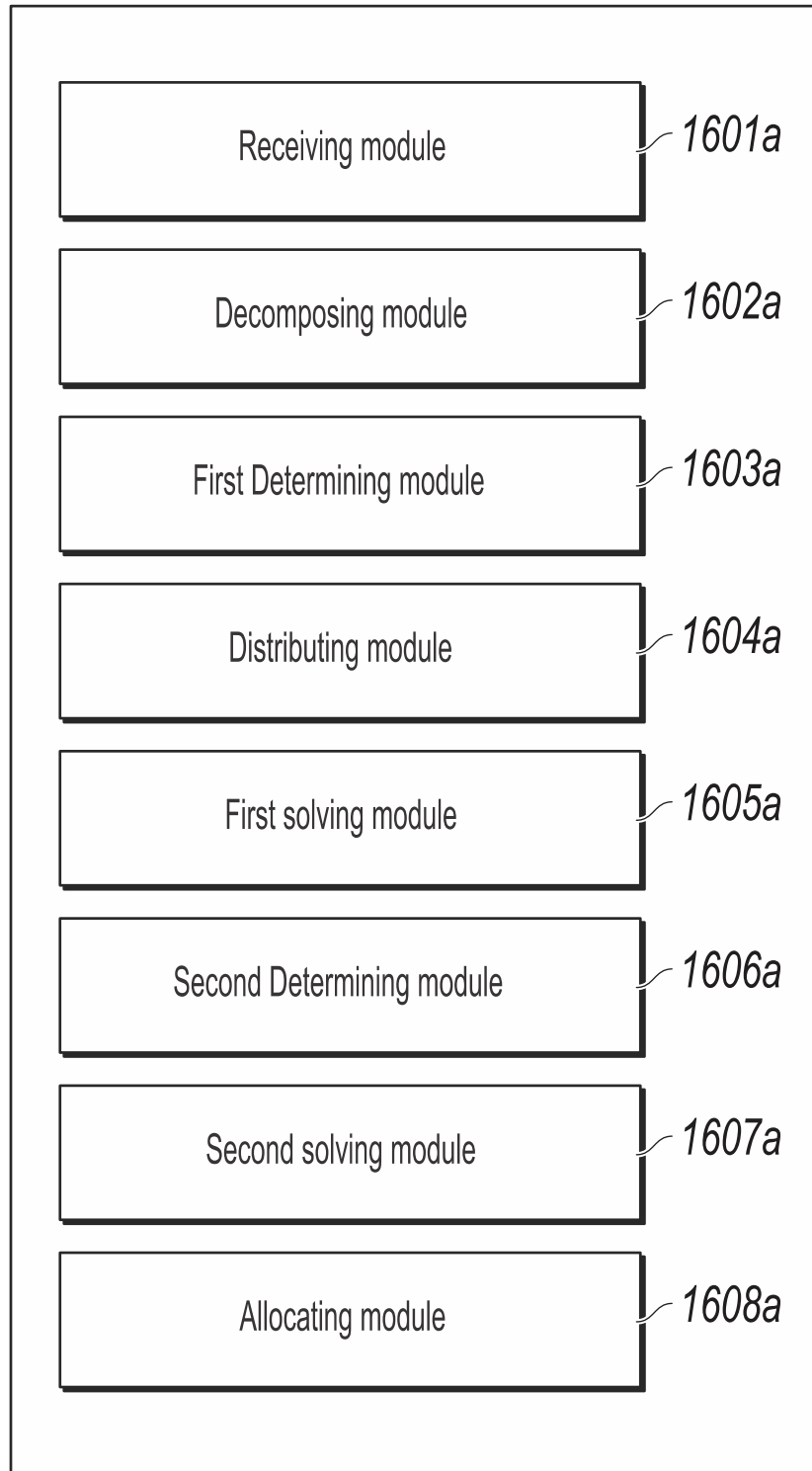
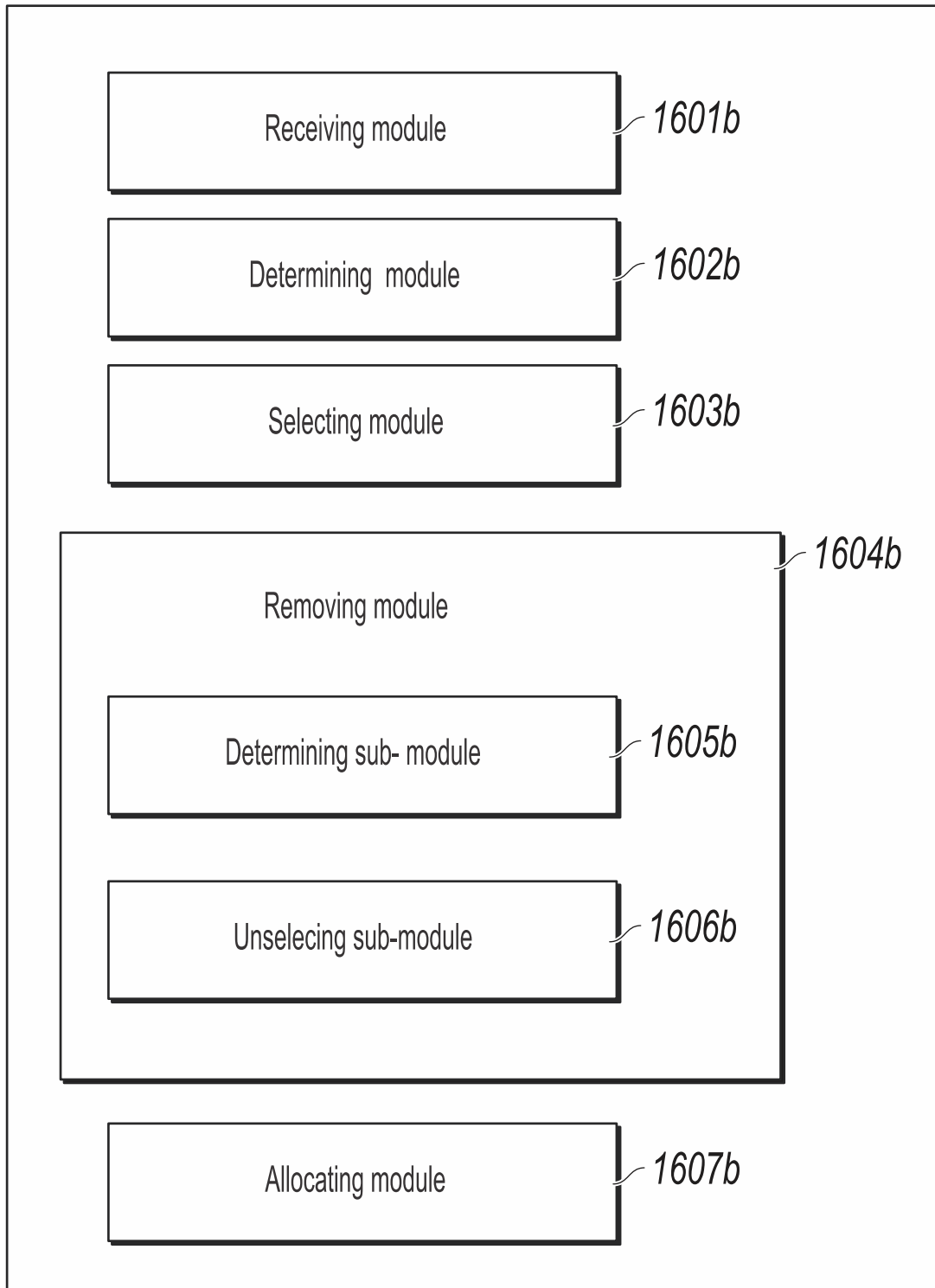


FIG. 15

**FIG. 16A**

**FIG. 16B**

1600c

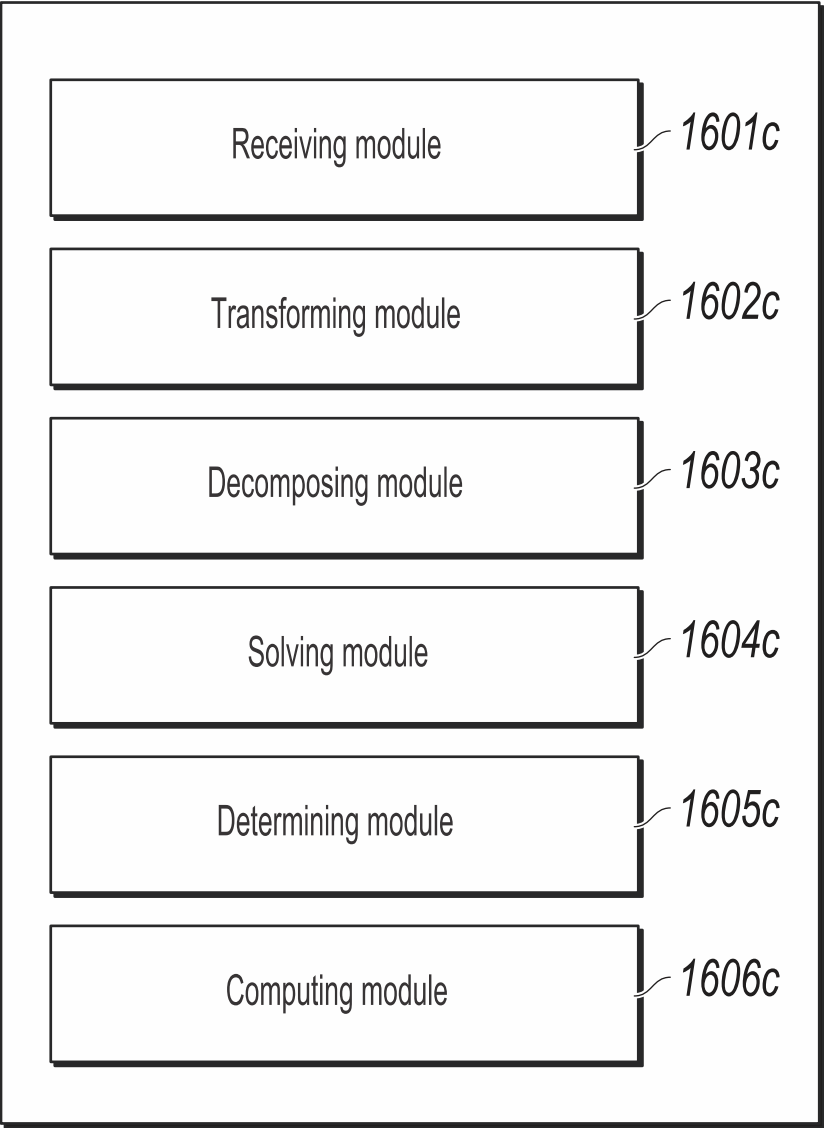


FIG. 16C