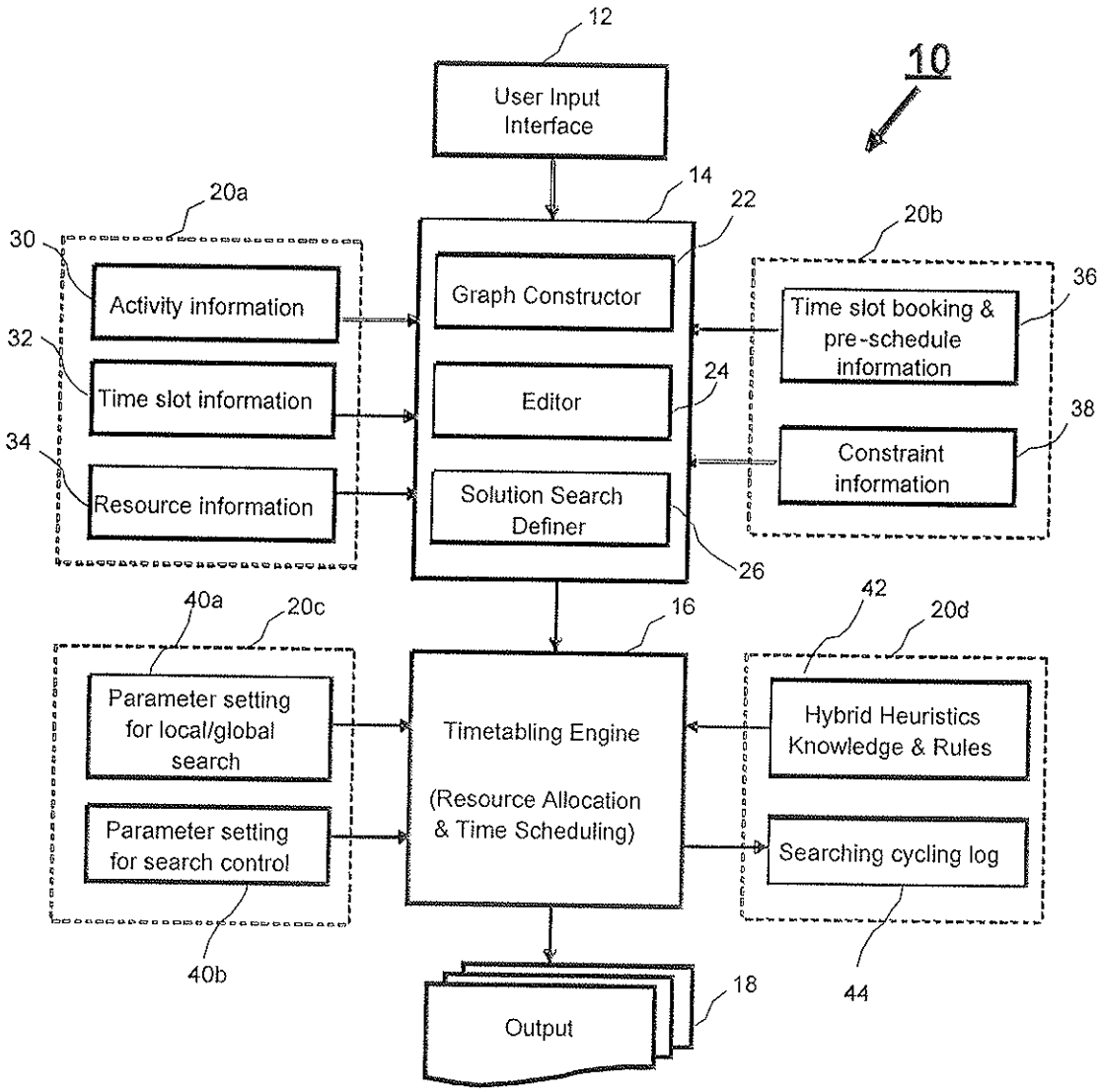
12
10
User Input Interface
14
22
20a
20b
30
Activity information
Graph Constructor
Time slot booking & pre-schedule information
36
32
Time slot information
Editor
24
38
34
Resource information
Constraint information
Solution Search Definer
26
40a
20c
16
42
20d
Parameter setting for local/global search
Timetabling Engine
(Resource Allocation & Time Scheduling)
Hybrid Heuristics Knowledge & Rules
Parameter setting for search control
Searching cycling log
40b
44
18
Output

FIG. 1

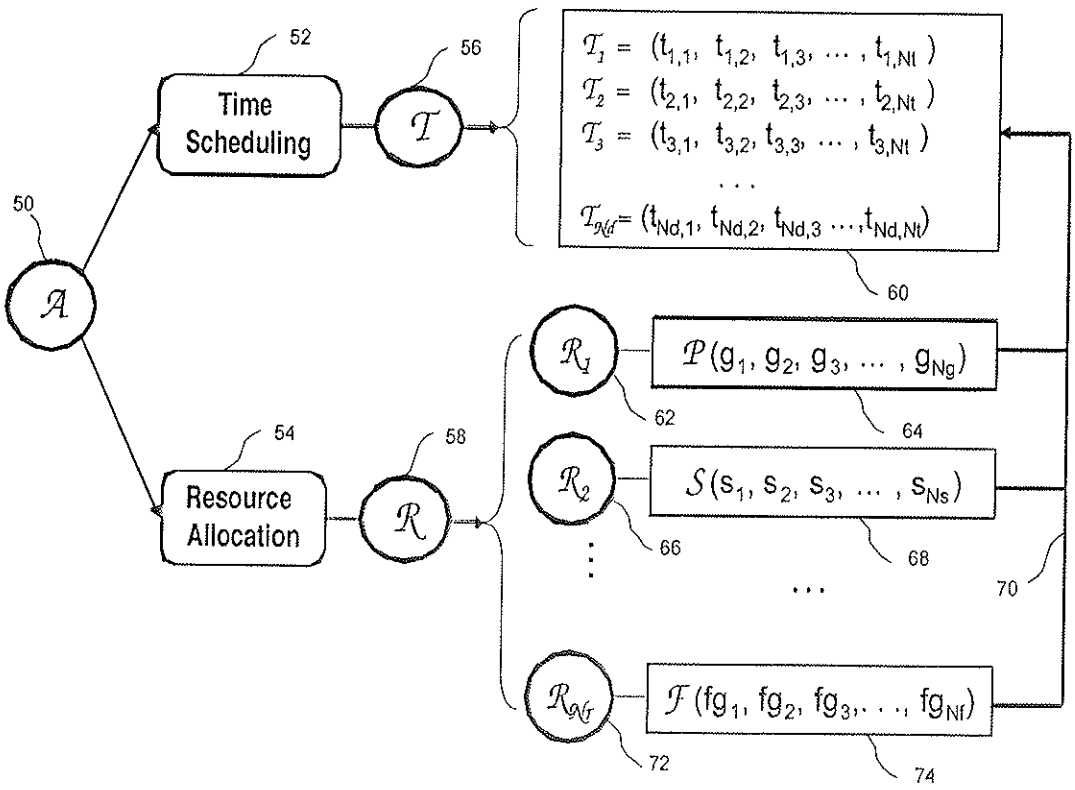
50
52
Time Scheduling
54
Resource Allocation
56
58
$\mathcal{T}_1 = (t_{1,1}, t_{1,2}, t_{1,3}, \ldots, t_{1,Nt})$
$\mathcal{T}_2 = (t_{2,1}, t_{2,2}, t_{2,3}, \ldots, t_{2,Nt})$
$\mathcal{T}_3 = (t_{3,1}, t_{3,2}, t_{3,3}, \ldots, t_{3,Nt})$
$\mathcal{T}_{Nd} = (t_{Nd,1}, t_{Nd,2}, t_{Nd,3} \ldots, t_{Nd,Nt})$
60
$\mathcal{P}(g_1, g_2, g_3, \ldots, g_{Ng})$
62
64
$\mathcal{S}(s_1, s_2, s_3, \ldots, s_{Ns})$
66
68
70
$\mathcal{F}(fg_1, fg_2, fg_3, \ldots, fg_{Nf})$
72
74

FIG. 2

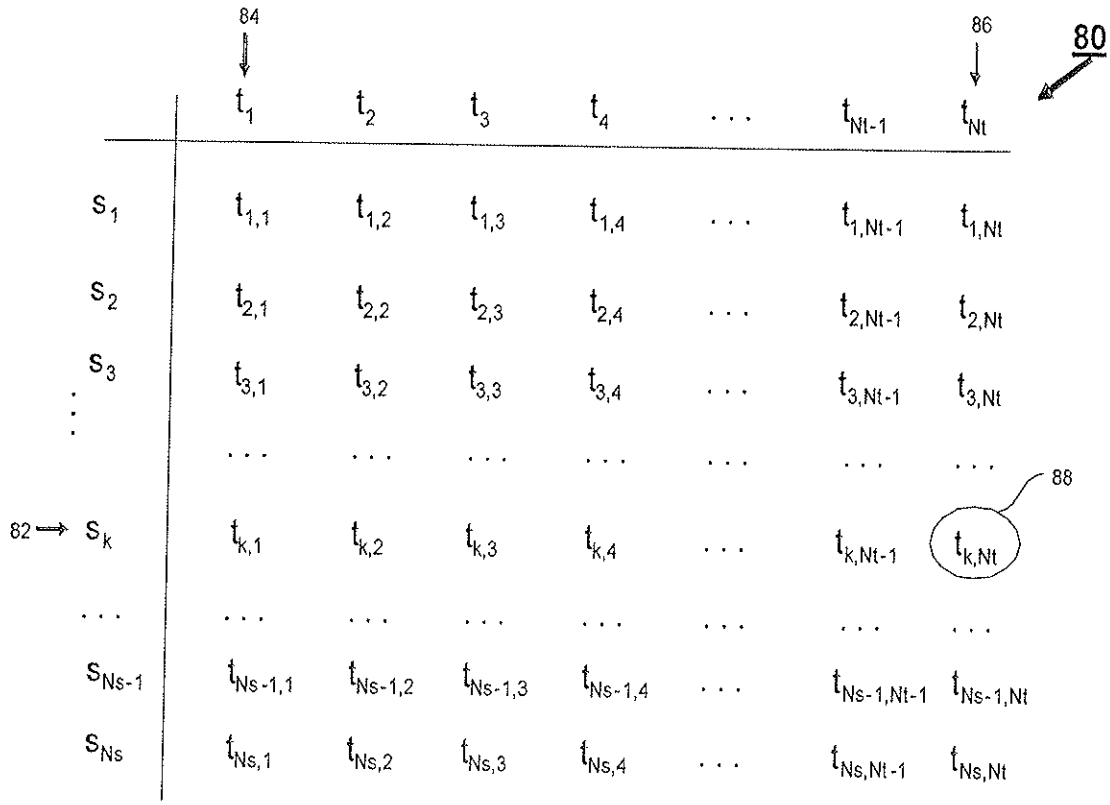
84
86
80
88
82
t_1 t_2 t_3 t_4 . . . t_{Nt-1} t_{Nt}
s_1 $t_{1,1}$ $t_{1,2}$ $t_{1,3}$ $t_{1,4}$ . . . $t_{1,Nt-1}$ $t_{1,Nt}$
s_2 $t_{2,1}$ $t_{2,2}$ $t_{2,3}$ $t_{2,4}$ . . . $t_{2,Nt-1}$ $t_{2,Nt}$
s_3 $t_{3,1}$ $t_{3,2}$ $t_{3,3}$ $t_{3,4}$ . . . $t_{3,Nt-1}$ $t_{3,Nt}$
s_k $t_{k,1}$ $t_{k,2}$ $t_{k,3}$ $t_{k,4}$ . . . $t_{k,Nt-1}$ $t_{k,Nt}$
s_{Ns-1} $t_{Ns-1,1}$ $t_{Ns-1,2}$ $t_{Ns-1,3}$ $t_{Ns-1,4}$ . . . $t_{Ns-1,Nt-1}$ $t_{Ns-1,Nt}$
s_{Ns} $t_{Ns,1}$ $t_{Ns,2}$ $t_{Ns,3}$ $t_{Ns,4}$ . . . $t_{Ns,Nt-1}$ $t_{Ns,Nt}$

FIG. 3

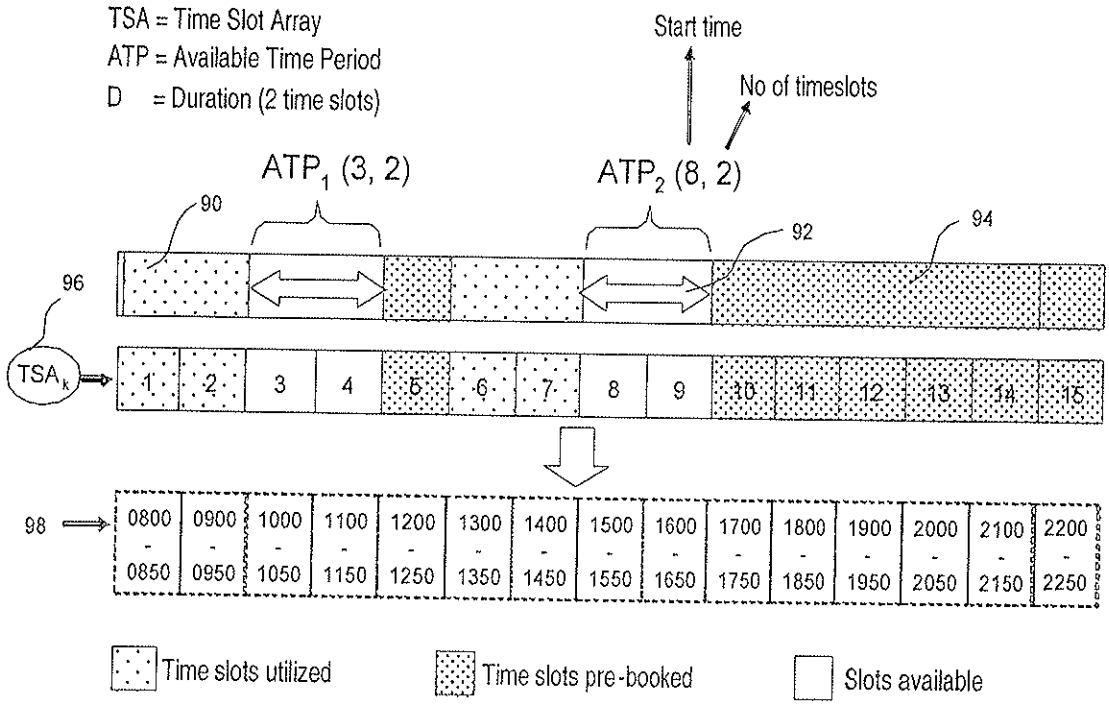
TSA = Time Slot Array
ATP = Available Time Period
D = Duration (2 time slots)
Start time
No of timeslots
ATP1 (3, 2)
ATP2 (8, 2)
90
92
94
96
TSAk
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
98
0800 - 0850
0900 - 0950
1000 - 1050
1100 - 1150
1200 - 1250
1300 - 1350
1400 - 1450
1500 - 1550
1600 - 1650
1700 - 1750
1800 - 1850
1900 - 1950
2000 - 2050
2100 - 2150
2200 - 2250
Time slots utilized
Time slots pre-booked
Slots available

FIG. 4

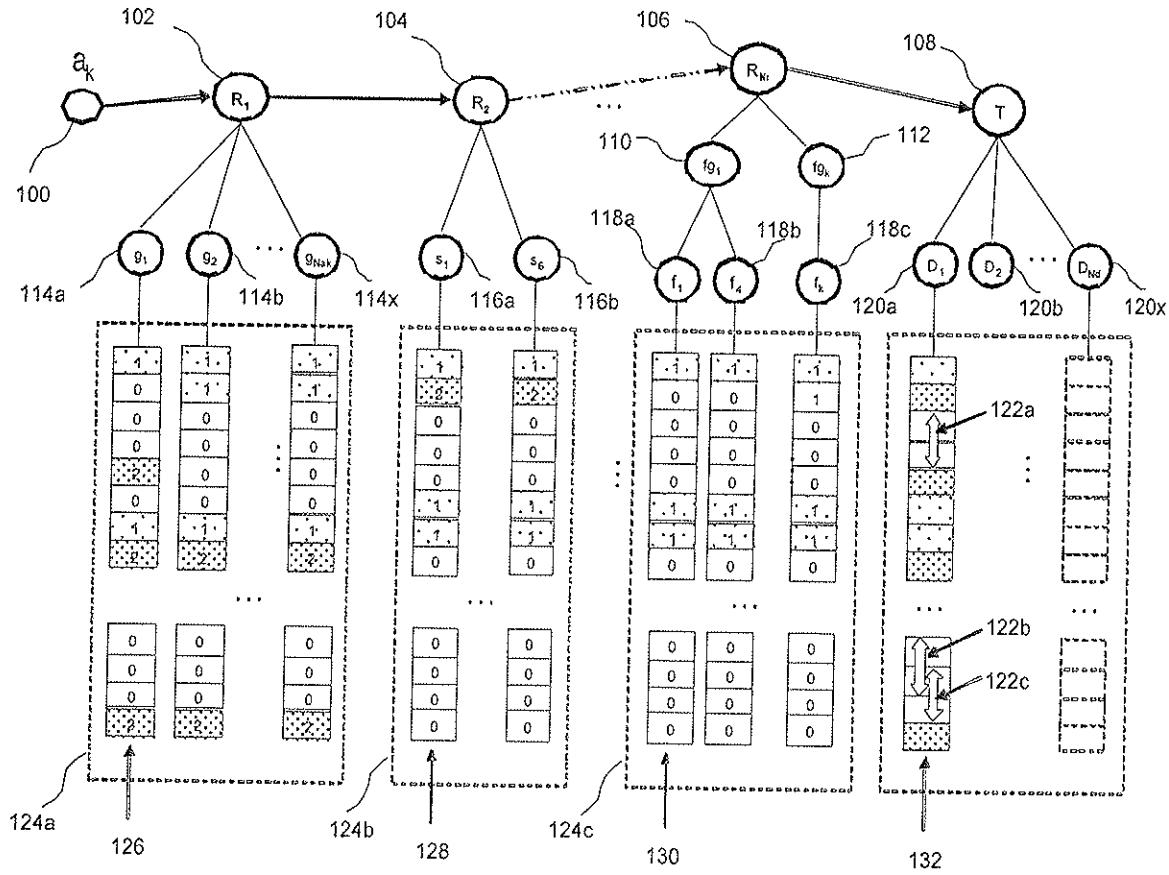
102
104
106
108
a_k
R_1
R_2
R_{Nr}
T
100
110
112
fg_1
fg_k
118a
118b
118c
g_1
g_2
g_{Nak}
s_1
s_6
f_1
f_4
f_k
D_1
D_2
D_{Nd}
114a
114b
114x
116a
116b
120a
120b
120x
122a
122b
122c
124a
124b
124c
126
128
130
132

FIG. 5

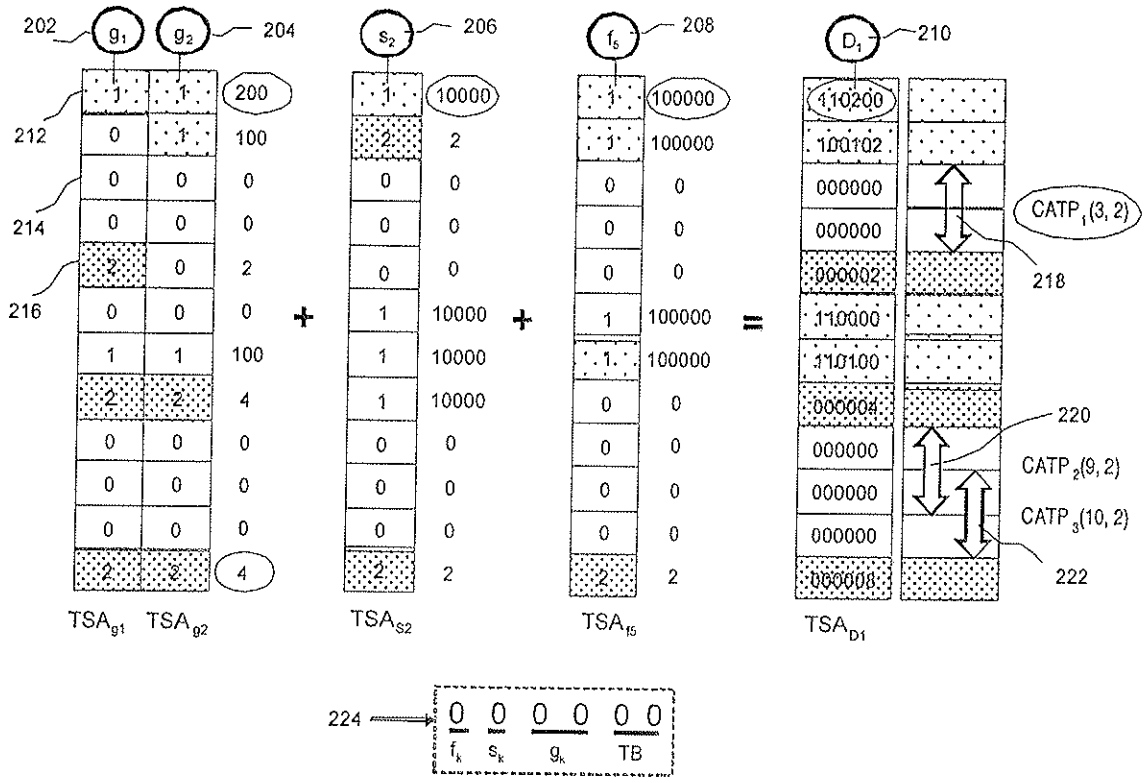
202
g1
g2
204
200
212
214
216
100
4
TSAg1
TSAg2
s2
206
10000
TSAS2
f5
208
100000
TSAf5
D1
210
TSAD1
CATP1(3, 2)
218
220
CATP2(9, 2)
CATP3(10, 2)
222
224
0 0 0 0 0 0
fk sk gk TB

FIG. 6

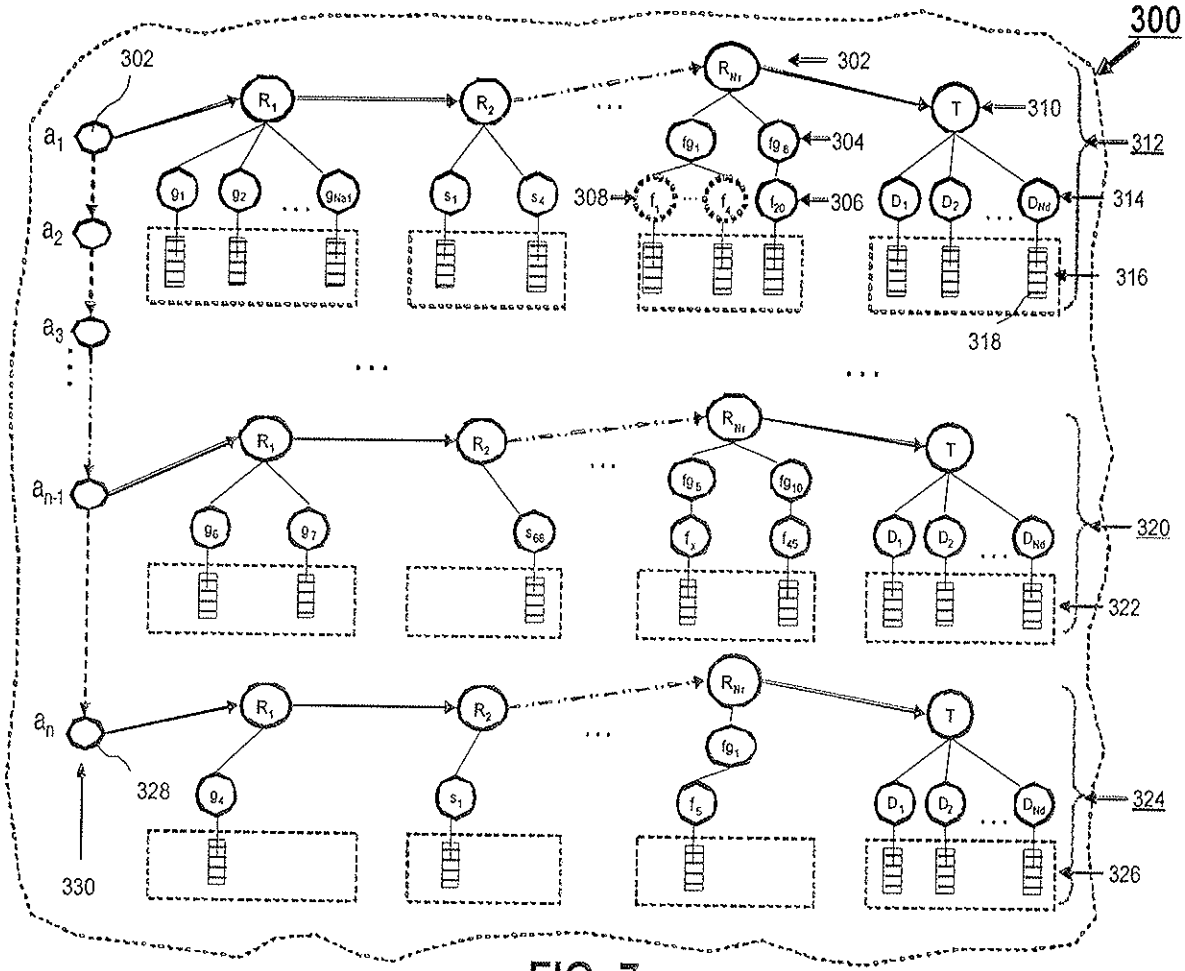
300
302
302
310
312
304
306
308
314
316
318
320
322
324
326
328
330
a_1
a_2
a_3
a_{n-1}
a_n
R_1
R_2
R_{N1}
T
D_1
D_2
D_{Nd}

FIG. 7

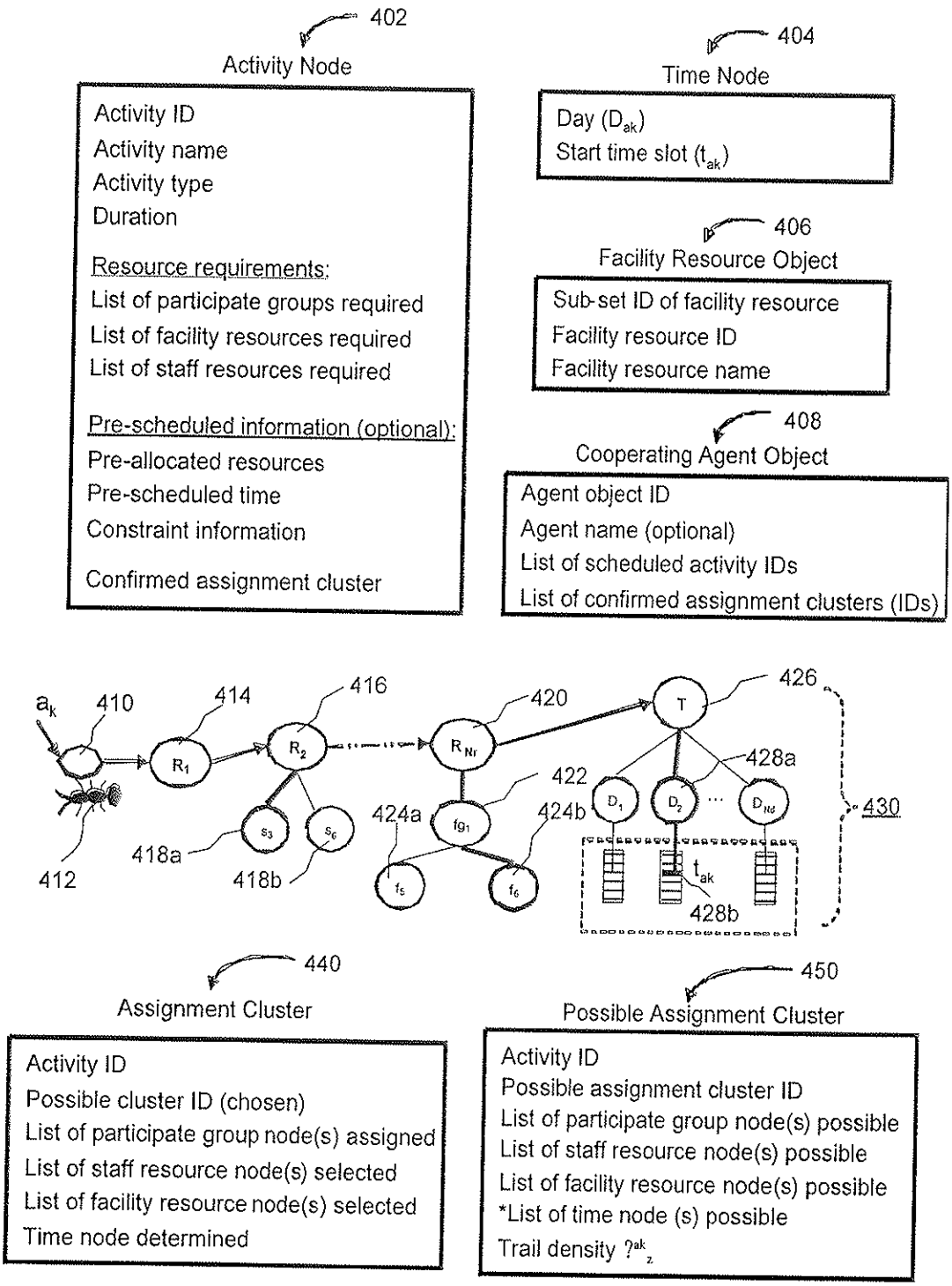
402
Activity Node
Activity ID
Activity name
Activity type
Duration
Resource requirements:
List of participate groups required
List of facility resources required
List of staff resources required
Pre-scheduled information (optional):
Pre-allocated resources
Pre-scheduled time
Constraint information
Confirmed assignment cluster
404
Time Node
Day (D_{ak})
Start time slot (t_{ak})
406
Facility Resource Object
Sub-set ID of facility resource
Facility resource ID
Facility resource name
408
Cooperating Agent Object
Agent object ID
Agent name (optional)
List of scheduled activity IDs
List of confirmed assignment clusters (IDs)
a_k
410
412
414
416
418a
418b
420
422
424a
424b
426
428a
428b
430
440
Assignment Cluster
Activity ID
Possible cluster ID (chosen)
List of participate group node(s) assigned
List of staff resource node(s) selected
List of facility resource node(s) selected
Time node determined
450
Possible Assignment Cluster
Activity ID
Possible assignment cluster ID
List of participate group node(s) possible
List of staff resource node(s) possible
List of facility resource node(s) possible
*List of time node (s) possible
Trail density ?$^{ak}_{z}$

FIG. 8

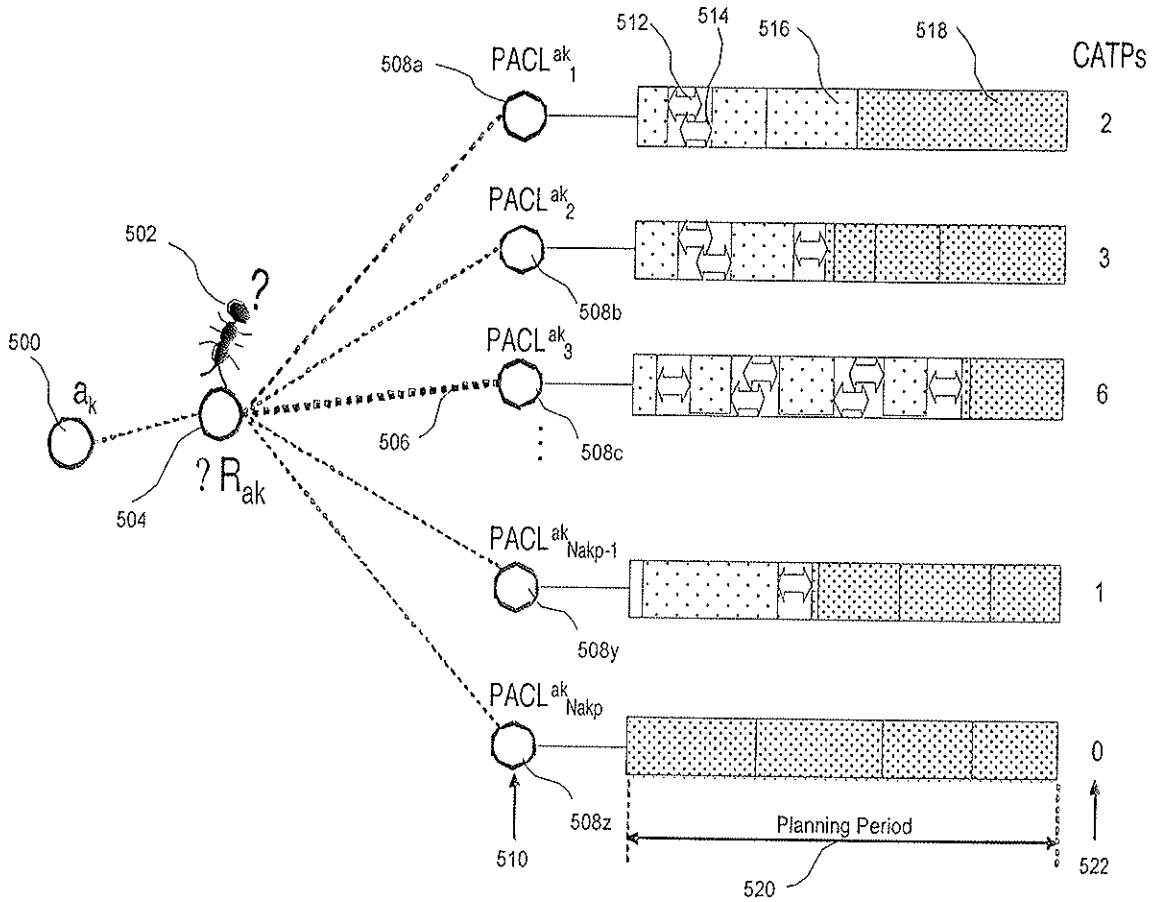
512
514
516
518
CATPs
508a
PACL ak 1
2
PACL ak 2
3
502
?
508b
500
ak
PACL ak 3
6
506
508c
504
? Rak
PACL ak Nakp-1
1
508y
PACL ak Nakp
0
508z
Planning Period
510
520
522

FIG. 9

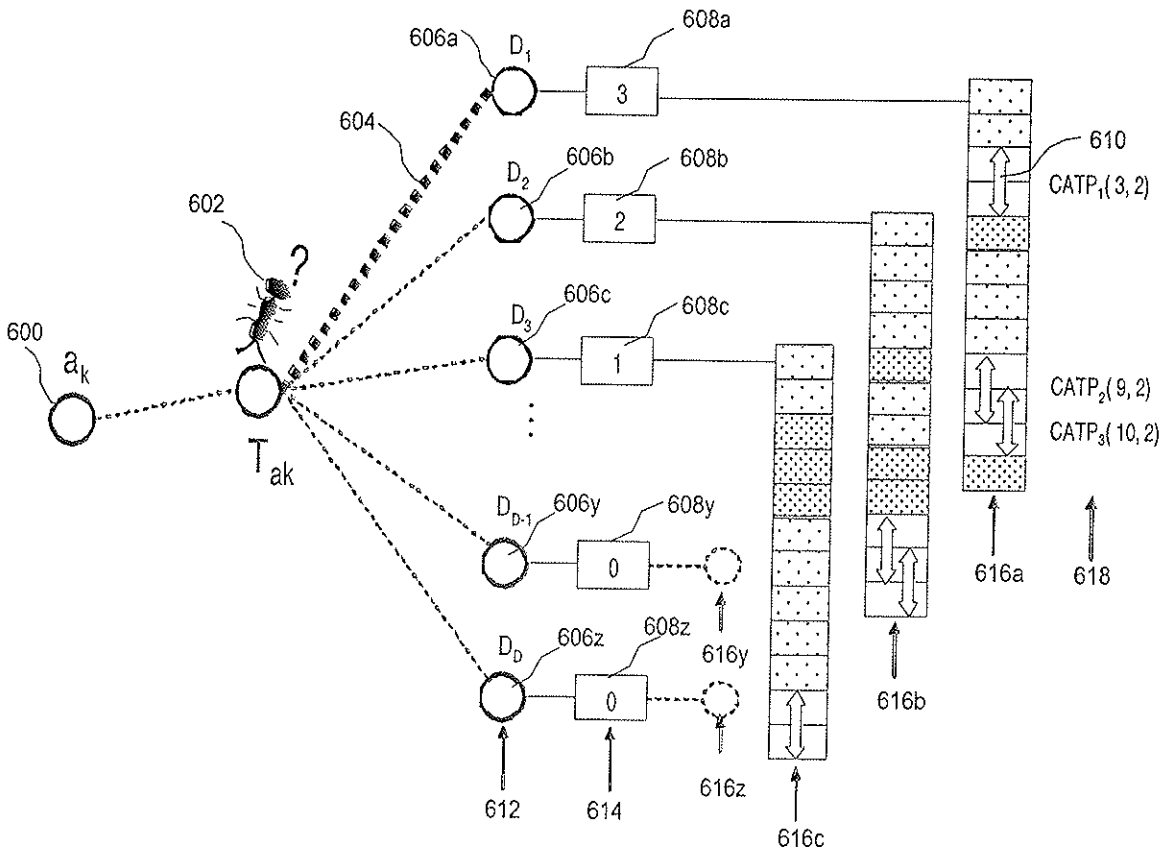
600
a_k
602
?
604
T_{ak}
606a
D_1
608a
3
606b
D_2
608b
2
606c
D_3
608c
1
D_{D-1}
606y
608y
0
616y
D_D
606z
608z
0
616z
612
614
610
$CATP_1(3,2)$
$CATP_2(9,2)$
$CATP_3(10,2)$
616a
618
616b
616c

FIG. 10

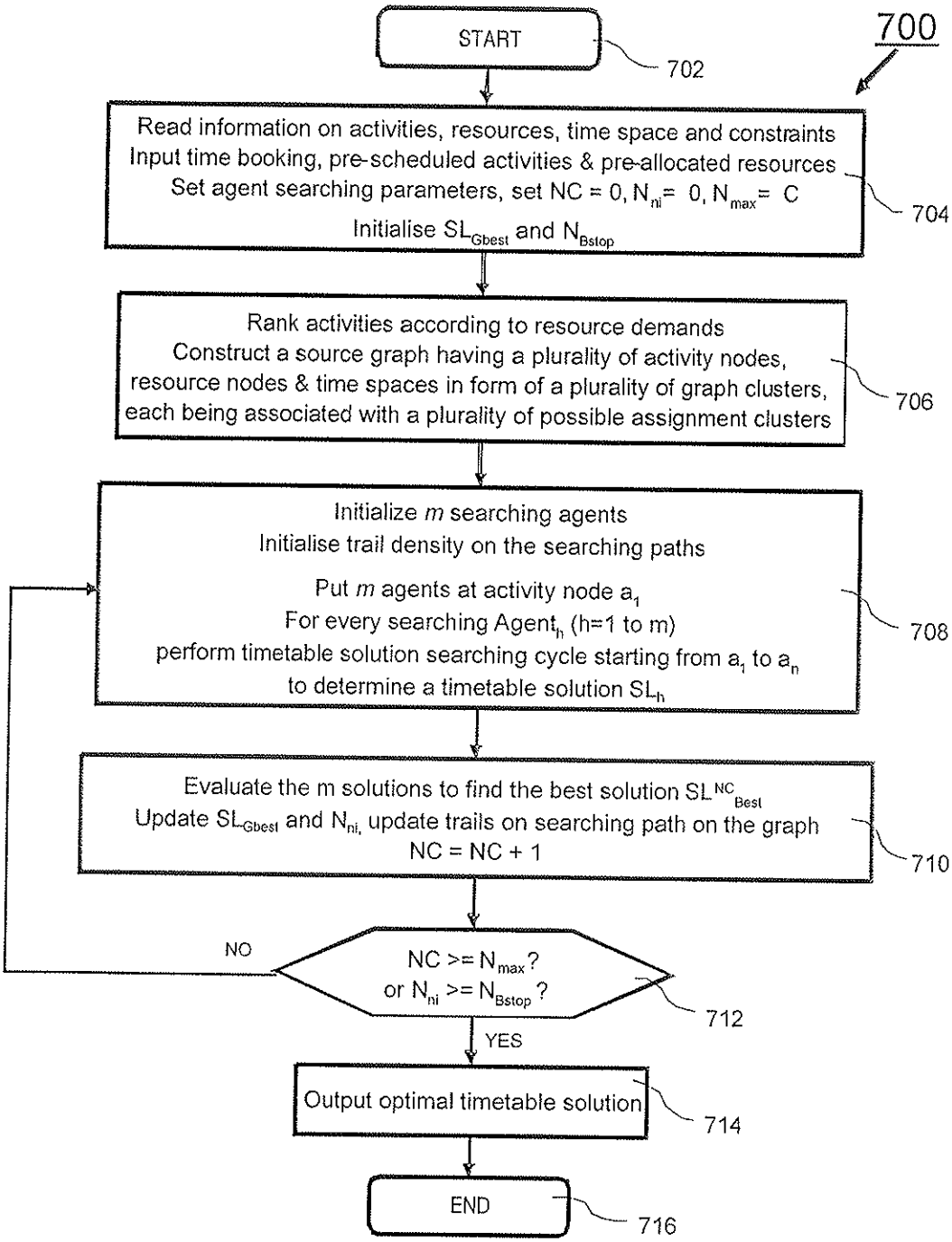
700
START
702
Read information on activities, resources, time space and constraints
Input time booking, pre-scheduled activities & pre-allocated resources
Set agent searching parameters, set NC = 0, N_{ni}= 0, N_{max}= C
Initialise SL_{Gbest} and N_{Bstop}
704
Rank activities according to resource demands
Construct a source graph having a plurality of activity nodes,
resource nodes & time spaces in form of a plurality of graph clusters,
each being associated with a plurality of possible assignment clusters
706
Initialize *m* searching agents
Initialise trail density on the searching paths
Put *m* agents at activity node a_1
For every searching $Agent_h$ (h=1 to m)
perform timetable solution searching cycle starting from a_1 to a_n
to determine a timetable solution SL_h
708
Evaluate the m solutions to find the best solution SL^{NC}_{Best}
Update SL_{Gbest} and $N_{ni,}$ update trails on searching path on the graph
NC = NC + 1
710
NO
NC >= N_{max} ?
or N_{ni} >= N_{Bstop} ?
712
YES
Output optimal timetable solution
714
END
716

FIG. 11

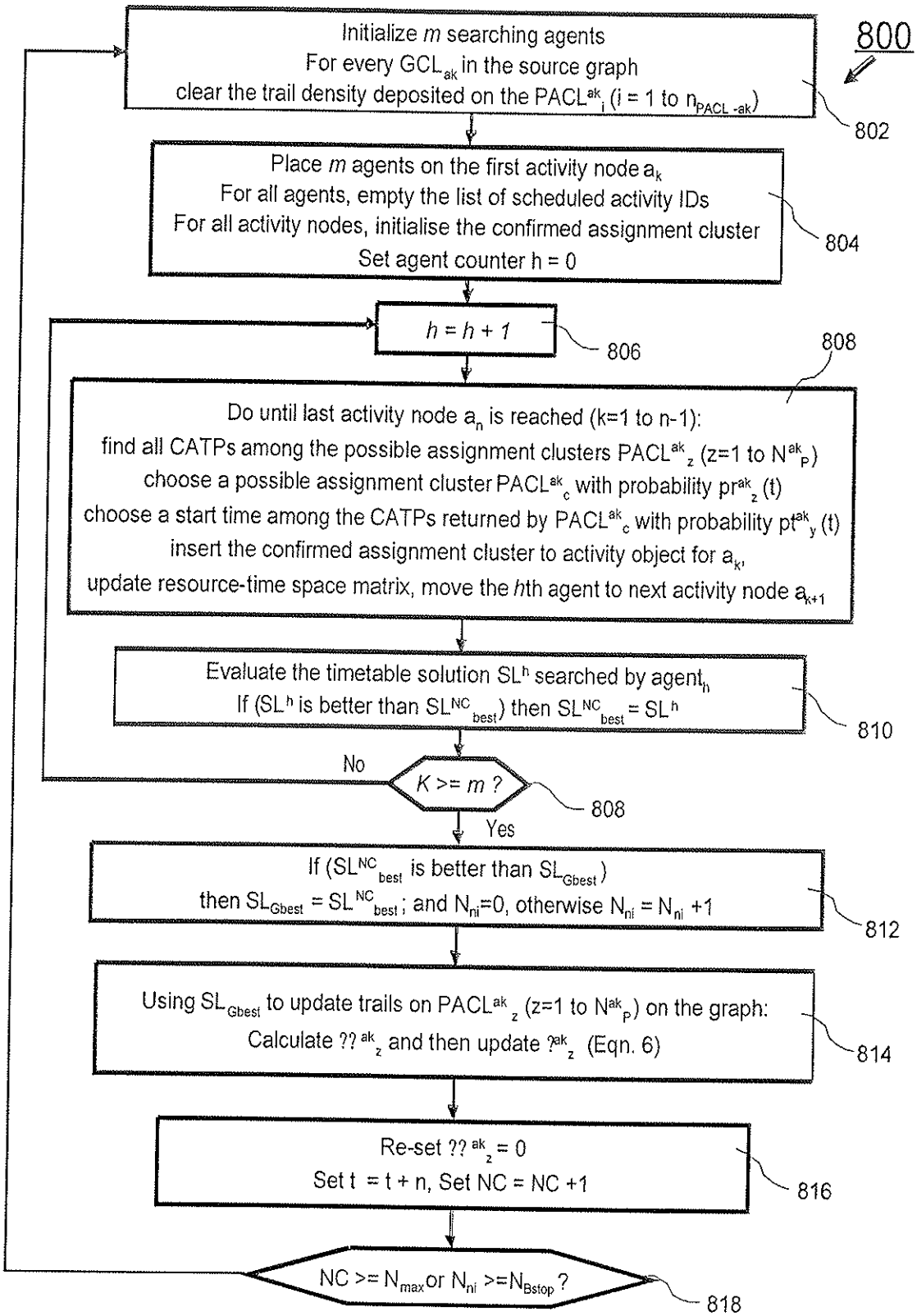
800
Initialize m searching agents
For every GCL_ak in the source graph
clear the trail density deposited on the PACL^ak_i (i = 1 to n_PACL -ak)
802
Place m agents on the first activity node a_k
For all agents, empty the list of scheduled activity IDs
For all activity nodes, initialise the confirmed assignment cluster
Set agent counter h = 0
804
h = h + 1
806
808
Do until last activity node a_n is reached (k=1 to n-1):
find all CATPs among the possible assignment clusters PACL^ak_z (z=1 to N^ak_P)
choose a possible assignment cluster PACL^ak_c with probability pr^ak_z (t)
choose a start time among the CATPs returned by PACL^ak_c with probability pt^ak_y (t)
insert the confirmed assignment cluster to activity object for a_k,
update resource-time space matrix, move the hth agent to next activity node a_k+1
Evaluate the timetable solution SL^h searched by agent_h
If (SL^h is better than SL^NC_best) then SL^NC_best = SL^h
810
No
K >= m ?
808
Yes
If (SL^NC_best is better than SL_Gbest)
then SL_Gbest = SL^NC_best; and N_ni=0, otherwise N_ni = N_ni +1
812
Using SL_Gbest to update trails on PACL^ak_z (z=1 to N^ak_P) on the graph:
Calculate ?? ^ak_z and then update ?^ak_z (Eqn. 6)
814
Re-set ?? ^ak_z = 0
Set t = t + n, Set NC = NC +1
816
NC >= N_max or N_ni >=N_Bstop ?
818

FIG. 12

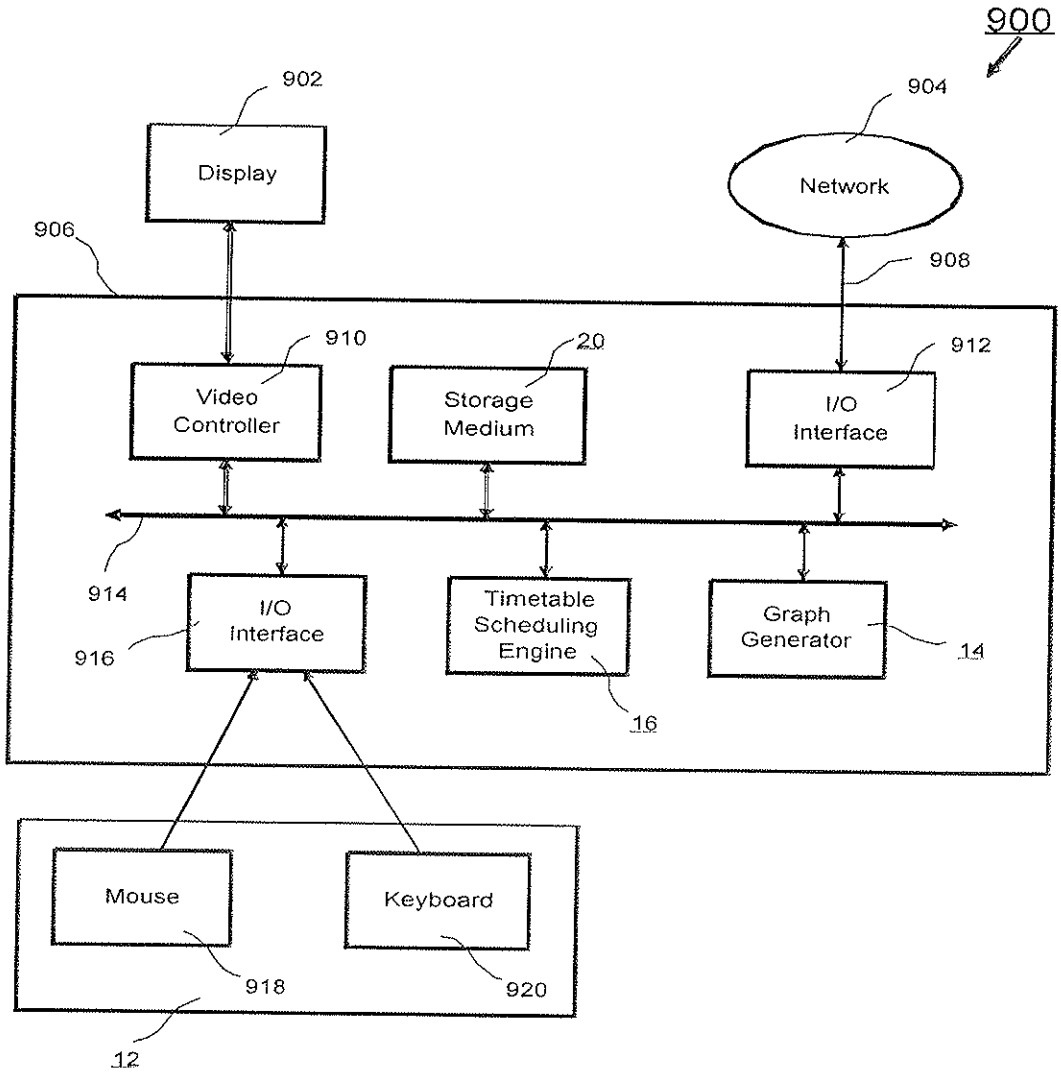
900
902
Display
904
Network
906
908
910
Video Controller
20
Storage Medium
912
I/O Interface
914
916
I/O Interface
Timetable Scheduling Engine
16
Graph Generator
14
Mouse
Keyboard
918
920
12

FIG. 13