**(1)** The TSS-based BFS method for optimizing the makespan of AMSs is given in Algorithm 1.

|  |
| --- |
| **Algorithm 1** TSS-based BFS for optimizing the makespan of AMSs |
| Input: An AMS and its PPN model.  Output: Optimal schedule α and *Cmax*(α). |
| 1: Initialize: OPEN[0] = {*TM*0[ε]}, OPEN[*i*] = ∅, *i* ∈ *ZK*, and FINAL = ∅. /\* OPEN[*i*], *i* ∈ *ZK* is a list used for storing timed states generated in layer *i* when searching the TSS of an AMS, *K* is the maximum layer value in the TSS, and FINAL is a list used for storing final timed markings. \*/  2: *k* = 0;  3: **while**(OPEN[*k*] ≠ ∅)do{  4: **for**(*TM*[υ] ∈ OPEN[*k*]){  5: Compute Δ(*TM*[υ]);/\*Δ(*TM*[υ]) is a set of transitions that are enabled at *TM*[υ]. \*/  6: **for**(*t* ∈ Δ(*TM*[υ])){  7: Fire transition *t*, obtain υ1 = υ*t* and *TM*1[υ1];  8: Δ(*TM*[υ]) := Δ(*TM*[υ])\*t*;  9: if(*TM*1[υ1] is a new final timed state){FINAL := FINAL∪*TM*1[υ1];}  10: else if(there exist a timed state *TMf*[α] in FINAL satisfying *TMf*[α] = *TM*1[υ1]){  11: if(τ|α| < τ|υ1|){*TMf*[α]:= *TM*1[υ1];}}  12: else if(there exist a timed state *TM*2[υ2] in OPEN[*k*+1] satisfying *TM*2[υ2] = *TM*1[υ1]){  13: if(τ|υ2| < τ|υ1|){*TM*2[υ2] := *TM*1[υ1];}}  14: else{OPEN[*k*+1] := OPEN[*k*+1]∪*TM*1[υ1];} /\* *TM*1[υ1] is not in OPEN[*k*+1]. \*/  15: }**end for**  16: OPEN[*k*] := OPEN[*k*]\*TM*[υ];  17: }**end for**  18: *k* := *k* + 1;  19: }**end while**  20: Output the best schedule in FINAL;  21: **End** |

**(2)** The TSS-based A\* method for optimizing the makespan of AMSs is shown in Algorithm 2.

|  |
| --- |
| **Algorithm 2** TSS-based A\* algorithm for optimizing the makespan of AMSs |
| Input: An AMS and its PPN model.  Output: Optimal schedule α and *Cmax*(α). |
| 1: Initialize: OPEN = {*TM*0[ε]}and CLOSED = ∅. /\*OPEN is a list used for storing unexplored timed states, and CLOSED is a list used for storing explored timed states. \*/  2: *k* = 0;  3: **while**(OPEN ≠ ∅)do{  4: select the state *TM*[υ] with the smallest *f*(*TM*[υ]);/\**f*(*TM*[υ]) is the heuristic function.\*/  5: if(*TM*[υ] is the final timed marking){α := υ, *Cmax*(α) = τ|υ|, break;}  6: else{  7: Compute Δ(*TM*[υ]);/\*Δ(*TM*[υ]) is a set of transitions that are enabled at *TM*[υ]. \*/  8: **for**(*t* ∈ Δ(*TM*[υ])){  9: Fire transition *t*, obtain υ1 = υ*t* and *TM*1[υ1];  10: Δ(*TM*[υ]) := Δ(*TM*[υ])\*t*;  11: if(there exist a timed state *TM*2[υ2] in OPEN satisfying *TM*2[υ2] = *TM*1[υ1]){  12: if(τ|υ2| < τ|υ1|){*TM*2[υ2] := *TM*1[υ1];}}  13: else if(there exist a timed state *TM*3[υ3] in CLOSED satisfying *TM*3[υ3] = *TM*1[υ1]){  14: if(τ|υ3| < τ|υ1|){CLOSED := CLOSED\*TM*3[υ3]; OPEN := OPEN∪*TM*3[υ3];}}  15: else{OPEN := OPEN∪*TM*1[υ1];}/\* *TM*1[υ1] is neither in OPEN nor CLOSED. \*/  16: }**end for**  17: OPEN := OPEN\*TM*[υ];  18: CLOSED := CLOSED∪*TM*[υ];  19: }**end while**  20: Output α and *Cmax*(α);  21: **End** |

**(3)** The implementation of TSS-based A\* method is illustrated in Example 1.

*Example 1:* Consider the AMS in Example 2. Let the heuristic function in A\* be *f*(*TM*[υ]) = τ|υ|(υ), i.e., the actual time cost function of *TM*[υ], which is admissible. Fig. 1 records the first four search steps of A\* on the TSS of the considered AMS.

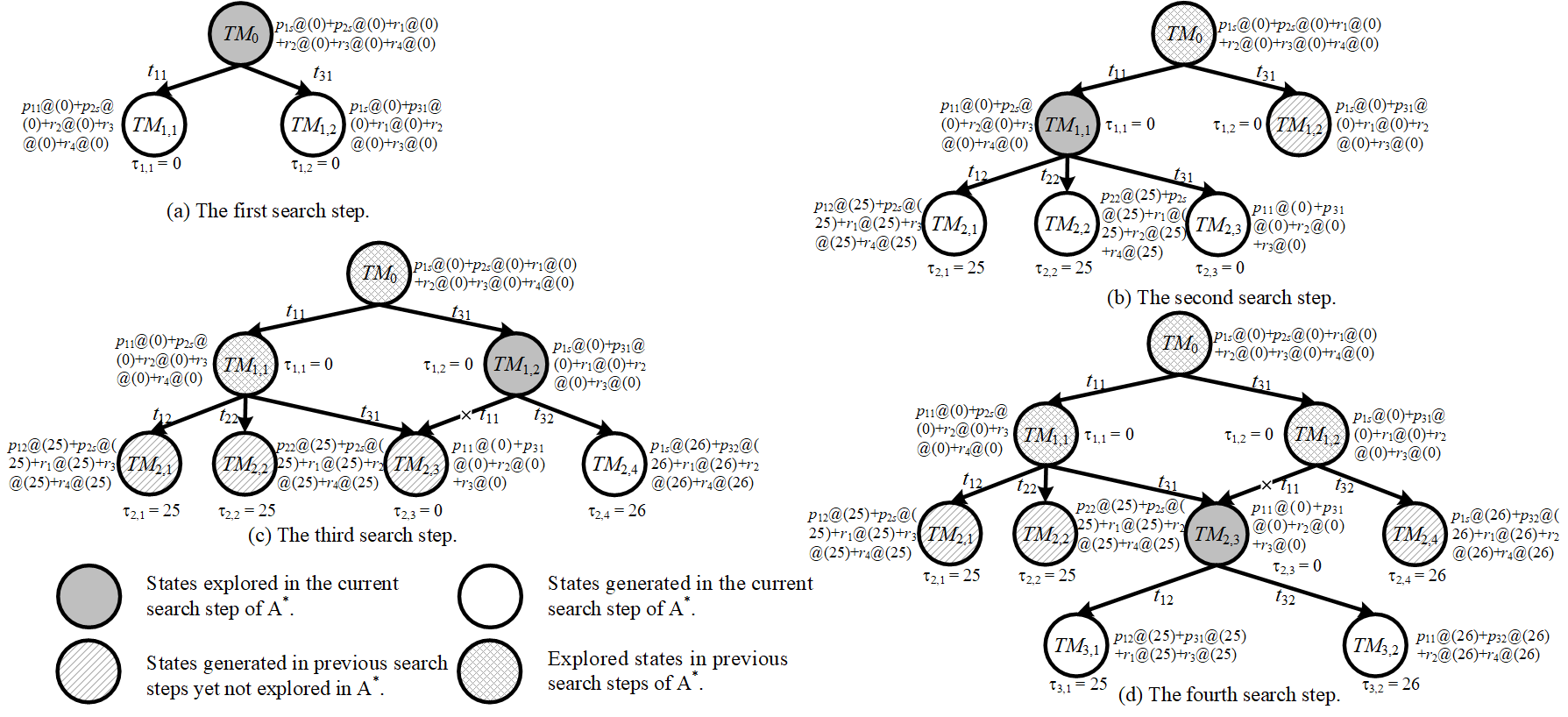


Fig. 1. The first four search steps of TSS-based A\*.

As shown in Fig. 1, the initial timed state *TM*0[ε] = *p*1*s*@(0) + *p*2*s*@(0) + *r*1@(0) + *r*2@(0) + *r*3@(0) + *r*4@(0) is searched in the first search step of A\*, and two new states *TM*1,1[υ1,1] and *TM*1,2[υ1,2] are generated. Since the heuristic function values of *TM*1,1[υ1,1] and *TM*1,2[υ1,2] are equal, i.e., *f*(*TM*1,1[υ1,1]) = *f*(*TM*1,2[υ1,2]) = 0, randomly select one for further exploration. In the second search step, *TM*1,1[υ1,1] is selected, and by separately firing the enabled transitions, three new states *TM*2,1[υ2,1], *TM*2,2[υ2,2], and *TM*2,3[υ2,3] can be generated in this step. Similarly, in the third and fourth search steps, the unexplored state with the best heuristic function value is selected for exploration, which are *TM*1,2[υ1,2] and *TM*2,3[υ2,3], respectively. In Fig. 1, the explored states, generated states, and generated but not explored states in the first four search steps of A\* are presented. ♣

**(4)** The four factor levels of each parameter are shown in Table I, and the RV values of HHS with different estimation functions are also recorded in Table I. Table II records the average RV (ARV) values of *w*0 and δ under different factor levels.

TABLE I

Parameter levels of *w*0 and δ, RV values of HHS under different parameter combinations

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Factor level | Parameter | |  | Experiment Number | Factor level | | RV | | |
| *w*0 | δ |  | *w*0 | δ | *f*1(*M*[υ]) | *f*2(*M*[υ]) | *f*3(*M*[υ]) |
| 1 | 10 | 1 |  | 1 | 1 | 1 | 241 | 226 | 234 |
| 2 | 20 | 2 |  | 2 | 1 | 2 | 230 | 233 | 230 |
| 3 | 30 | 3 |  | 3 | 1 | 3 | 242 | 223 | 231 |
| 4 | 40 | 4 |  | 4 | 1 | 4 | 227 | 224 | 224 |
|  |  |  |  | 5 | 2 | 1 | 234 | 230 | 230 |
|  |  |  |  | 6 | 2 | 2 | 235 | 231 | 230 |
|  |  |  |  | 7 | 2 | 3 | 231 | 232 | 237 |
|  |  |  |  | 8 | 2 | 4 | 235 | 236 | 231 |
|  |  |  |  | 9 | 3 | 1 | 238 | 228 | 238 |
|  |  |  |  | 10 | 3 | 2 | 236 | 241 | 237 |
|  |  |  |  | 11 | 3 | 3 | 238 | 242 | 235 |
|  |  |  |  | 12 | 3 | 4 | 238 | 243 | 230 |
|  |  |  |  | 13 | 4 | 1 | 233 | 232 | 227 |
|  |  |  |  | 14 | 4 | 2 | 237 | 232 | 227 |
|  |  |  |  | 15 | 4 | 3 | 230 | 231 | 225 |
|  |  |  |  | 16 | 4 | 4 | 233 | 232 | 233 |

Table II

ARV values of each parameter under different factor levels

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Factor level | *f*1(*M*[υ]) | |  | *f*2(*M*[υ]) | |  | *f*3(*M*[υ]) | |
| *w*0 | δ |  | *w*0 | δ |  | *w*0 | δ |
| 1 | 235 | 236.5 |  | **226.5** | **229** |  | 229.75 | 232.25 |
| 2 | 233.75 | 234.5 |  | 232.25 | 234.25 |  | 232 | 231 |
| 3 | 237.5 | 235.25 |  | 238.5 | 232 |  | 235 | 232 |
| 4 | **233.25** | **233.25** |  | 231.75 | 233.75 |  | **228** | **229.5** |

**(5)** The RPD value of each instance is recorded in Table III.

Table III

RPD values of HHS algorithm with and without DCP

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instance | *f*1(*M*[υ]) | |  | *f*2(*M*[υ]) | |  | *f*3(*M*[υ]) | |
| With DCP | Without DCP |  | With DCP | Without DCP |  | With DCP | Without DCP |
| In01 | 4.089% | 5.576% |  | 2.602% | 0.372% |  | **0.000%** | 0.743% |
| In02 | 7.337% | 3.804% |  | **0.000%** | 1.359% |  | 1.902% | 1.359% |
| In03 | 6.167% | 1.762% |  | **0.000%** | 1.542% |  | 1.322% | 0.441% |
| In04 | 10.261% | 5.037% |  | **0.000%** | 2.799% |  | 1.866% | 5.224% |
| In05 | 0.893% | 1.339% |  | **0.000%** | 1.786% |  | **0.000%** | 0.446% |
| In06 | 2.614% | 5.556% |  | 4.902% | 4.248% |  | **0.000%** | 1.634% |
| In07 | 2.688% | 1.882% |  | 2.957% | 1.613% |  | **0.000%** | 1.075% |
| In08 | **0.000%** | 3.401% |  | 2.268% | 3.855% |  | 0.454% | 0.680% |
| In09 | 0.606% | 1.818% |  | 3.030% | 3.636% |  | 1.212% | **0.000%** |
| In10 | 4.018% | 2.679% |  | 3.571% | 2.232% |  | **0.000%** | 2.232% |
| In11 | 1.832% | 3.297% |  | 2.930% | 0.733% |  | **0.000%** | 0.733% |
| In12 | **0.000%** | 0.309% |  | 0.309% | 1.543% |  | 0.617% | **0.000%** |
| In13 | **0.000%** | 6.494% |  | 2.597% | 1.299% |  | **0.000%** | 1.299% |
| In14 | 2.778% | 6.019% |  | 1.389% | 2.315% |  | 0.463% | **0.000%** |
| In15 | 1.969% | 1.575% |  | 2.362% | 3.937% |  | 0.787% | **0.000%** |
| In16 | 4.319% | 4.319% |  | **0.000%** | 0.332% |  | 0.332% | 2.326% |