Θέμα: Εύρεση Δικριτηριακά Βέλτιστων Διαδρομών

Νταλαγιώργος Αχιλλέας

Επιβλέπων: Σπυρίδων Κοντογιάννης

Πανεπιστήμιο Ιωαννίνων

Αντικείμενο εργασίας

ΠΡΟΒΛΗΜΑ

Εύρεση Δικριτηριακά Βέλτιστων Διαδρομών

NP-Hard
Δισεπίλυτο

ΛΥΣΗ

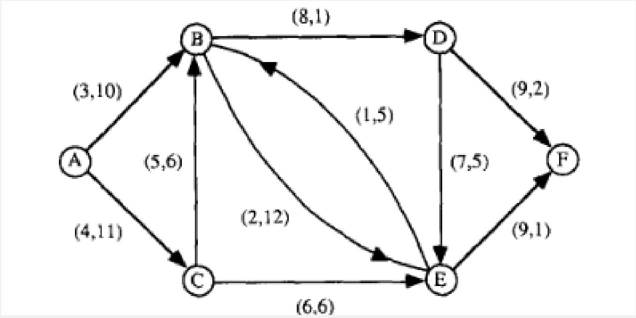
Αλγοριθμικές τεχνικές και προσεγγίσεις που έχουν αναπτυχθεί ανά τα χρόνια

4 Βασικές Προσεγγίσεις
- Label setting
- Label correcting
- Ranking method
- Two Phase Method

ΑΛΓΟΡΙΘΜΟΙ ΠΟΥ ΕΜΕΙΣ ΕΞΕΤΑΖΟΥΜΕ

(approximate) Bi-objective A (BOA*ε)*
Path Pair A (PPA*)*

Γράφημα με 2 κόστη κριτηρίων



Πρόβλημα

Εύρεση Διαδρομής που ελαχιστοποιεί και τα δύο κόστη κριτηρίων.

Λύση

Συνήθως δεν υπάρχει μια διαδρομή που ελαχιστοποιεί και τα δύο κόστη!

Οι αλγόριθμοι εύρεσης δικριτηριακά βέλτιστων διαδρομών παράγουν ως λύση Pareto βέλτιστο μέτωπο

*BOA*ε* και *PPA** υπολογίζουν το *κατά προσέγγιση Pareto βέλτιστο σύνολο.*

Βασικές Έννοιες

Κυριαρχία

- $c_1(\pi_v) < c_1(\pi'_v) \wedge c_2(\pi_v) < c_2(\pi'_v) \rightarrow \pi_v$ ***κυριαρχει αυστηρα*** π'_v
- $c_1(\pi_v) \leq c_1(\pi'_v) \wedge c_2(\pi_v) \leq c_2(\pi'_v) \rightarrow$

  π_v ***κυριαρχει ασθενως*** π'_v

Κατά Pareto βέλτιστο Μέτωπο Π_v

$if\ \pi\ \in$ Pareto frontier:

→ π δεν κυραρχείται αυστηρά από κανένα π′

$If\ \pi' \notin \Pi_v$

→ π′ κυριαρχείται ασθενώς από $\pi\ \in \Pi_v$

Κατά προσέγγιση Κυριαρχία

Έστω $\varepsilon_1, \varepsilon_2 \geq 0\ \in\ \Re : \pi_v$ ***(ε₁, ε₂)-κυριαρχεί*** π'_v

$c_1(\pi_v) \leq (1+\varepsilon_1) * c_1(\pi'_v) \wedge c_2(\pi_v) \leq (1+\varepsilon_2) * c_2(\pi'_v)$.

$\varepsilon_1, \varepsilon_2$ →*παράγοντες προσέγγισης.*

Κατά προσέγγιση Pareto Μέτωπο $\Pi_v^{(\varepsilon_1,\varepsilon_2)}$

- $\Pi_v^{(\varepsilon_1,\varepsilon_2)} \subseteq \Pi_v$.
- $\forall\ \pi \in \boldsymbol{\Pi_v^{(\varepsilon_1,\varepsilon_2)}}\ {}_{(\boldsymbol{\varepsilon_1,\varepsilon_2})}$***κυριαρχει*** $\forall\ \pi \in \Pi_v$

$BOA^*\varepsilon$ PPA^*

Οι αλγόριθμοι BOA* και PPA* αποτελούν πρόσφατες υλοποιήσεις που δίνουν λύση στο πρόβλημα εύρεσης Δικριτηριακά Βέλτιστων Διαδρομών.

ΕΙΣΟΔΟΣ

- $G=(V,E)$
- v_{start}, v_{goal}
- c_1, c_2
- $h_1, h_2 \rightarrow \min_{cost} \pi_v$ από v_{goal} προς καθε v .
- $\varepsilon_1, \varepsilon_2$

$BOA^*\varepsilon$ / PPA^* →

ΕΞΟΔΟΣ

Το κατά προσέγγιση Pareto βέλτιστο μέτωπο

Αρχικοποίηση 1η

*Παράδειγμα BOA*ε*

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

Ψευδοκώδικας BOA*ε

κόμβος x του BOA*

Αντιστοιχεί σε μία διαδρομή

$s(x), (g_1, g_2), (f_1, f_2), p(x)$

μία κορυφή v του G

κόστος διαδρομής

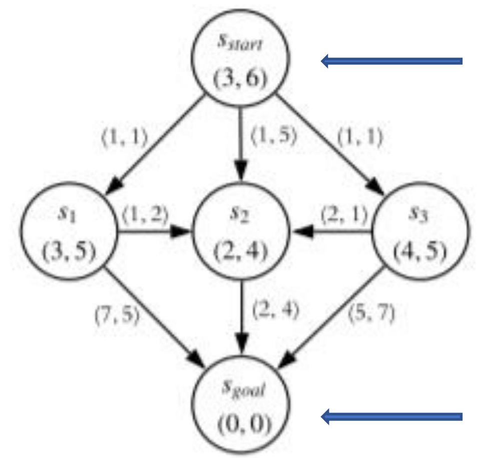
$f = g + h$ **κατώτερο όριο κόστους** πιθανών λύσεων.

η κορυφή πατέρας

Αρχικός κόμβος

$(s_{start}, (0,0), (3,6), null)$

κόμβος x του BOA* ≠ κορυφή v του γραφήματος G



Κορυφή εκκίνησης

Κορυφή στόχος

Αρχικοποίηση 2η

*Παράδειγμα BOA*ε*

Open

$(s_{start}, (0,0), (3,6), null)$

OpenList

Λίστα δυαδικού σωρού

$_{(f_1, f_2)}$*λεξικογραφική ταξινόμηση*

$$x \leq_{lex(f_1,f_2)} x' \Rightarrow x_{f1} < x'_{f_1} \vee \left(x_{f_1} = x'_{f_1} \wedge x_{f_2} < x'_{f_2}\right)$$

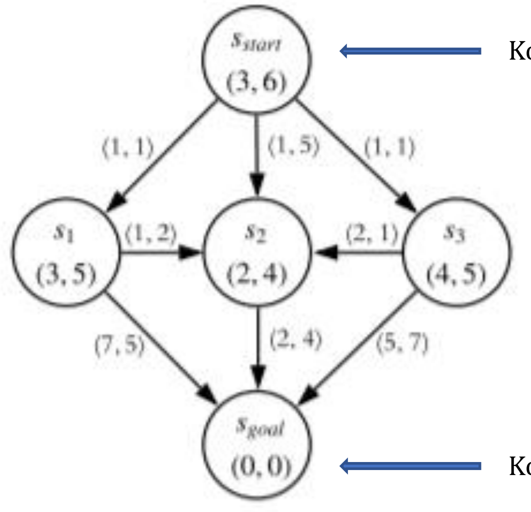
Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
     lexicographically smallest f-value of all nodes in
     Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*



Κορυφή εκκίνησης

Κορυφή στόχος

Αρχικοποίηση 3η

*Παράδειγμα BOA*ε*

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

Output: A cost-unique Pareto-optimal solution set

$sols \leftarrow \emptyset$
for each $s \in S$ **do**
  $g_2^{\min}(s) \leftarrow \infty$
$x \leftarrow$ new node with $s(x) = s_{start}$
$\mathbf{g}(x) \leftarrow (0, 0)$
$parent(x) \leftarrow$ null
$\mathbf{f}(x) \leftarrow (h_1(s_{start}), h_2(s_{start}))$
Initialize $Open$ and add x to it
while $Open \neq \emptyset$ **do**
  Remove a node x from $Open$ with the lexicographically smallest f-value of all nodes in $Open$
  if $g_2(x) \geq g_2^{\min}(s(x)) \vee (1+\varepsilon)*f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
    continue
  $g_2^{\min}(s(x)) \leftarrow g_2(x)$
  if $s(x) = s_{goal}$ **then**
    Add x to $sols$
    continue
  for each $t \in \mathrm{Succ}(s(x))$ **do**
    $y \leftarrow$ new node with $s(y) = t$
    $\mathbf{g}(y) \leftarrow \mathbf{g}(x) + \mathbf{c}(s(x), t)$
    $parent(y) \leftarrow x$
    $\mathbf{f}(y) \leftarrow \mathbf{g}(y) + \mathbf{h}(t)$
    if $g_2(y) \geq g_2^{\min}(t) \vee (1+\varepsilon)*f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
      continue
    Add y to $Open$
return $sols$

Ψευδοκώδικας BOA*ε

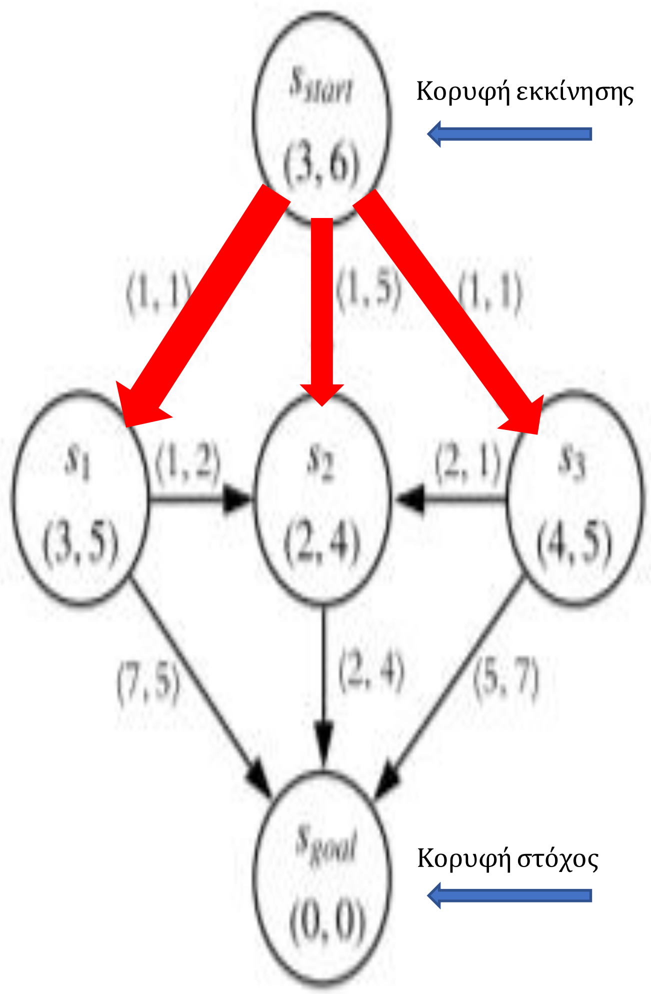
λίστα sols → *Λύσεις αλγορίθμου*

λίστα g_2^{min} → $min(g_2)\ \forall\ v$ → *Έλεγχο κυριαρχίας*

∞	∞	∞	∞	∞	g_2^{min}

Παράδειγμα BOA*ε=0

επανάληψη 1



<u>εκτέλεση</u>

$(s_{start}, (0,0), (3,6), null)$

0,6<∞ ->not
sstart-Sgoal
dominated

Open

$(s_{start}, (0,0), (3,6), null)$

$(s_2, (1,5), (3,9), s_{start})$ → 5 , 9 < ∞ ->not dominated

$(s_1, (1,1), (4,6), s_{start})$ → 1 , 6 < ∞ ->not dominated

$(s_3, (1,1), (5,6), s_{start})$ → 1 , 6 < ∞ ->not dominated

0	∞	∞	∞	∞	g_2^{min}

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

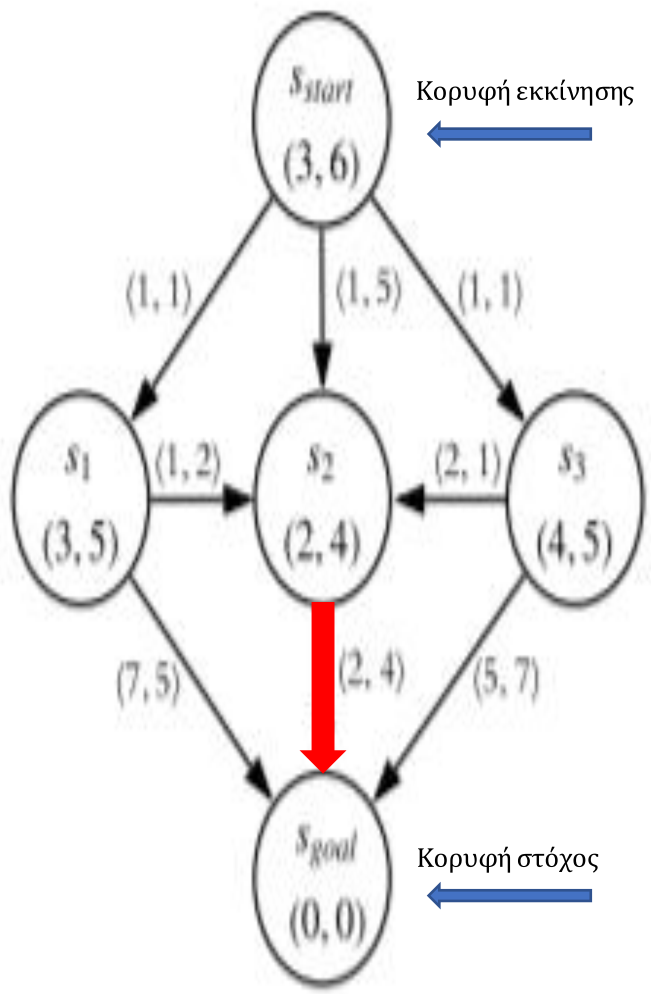
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

Παράδειγμα BOA*ε=0

επανάληψη 2



Κορυφή εκκίνησης

Κορυφή στόχος

$(s_2, (1,5), (3,9), s_{start})$ **-> not dominated**

Open

$(s_{goal}, (3,9), (3,9), s_2)$

$(s_1, (1,1), (4,6), s_{start})$

$(s_3, (1,1), (5,6), s_{start})$

0	∞	5	∞	∞	g_2^{min}

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

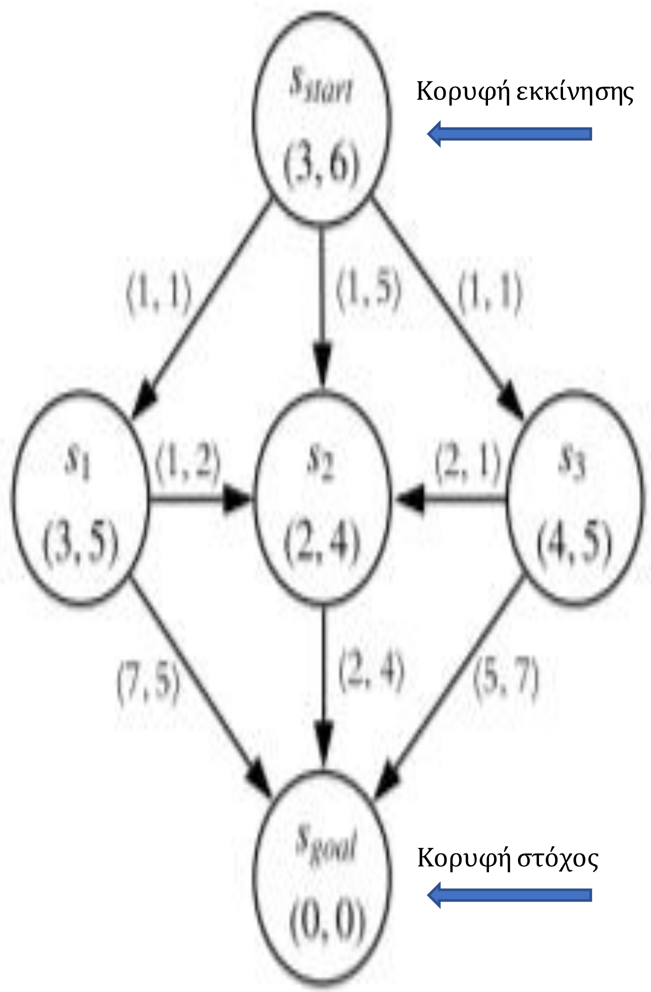
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2_min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2_min(s(x)) ∨ (1+ε)*f2(x) ≥ g2_min(s_goal) then
        continue
    g2_min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2_min(t) ∨ (1+ε)*f2(x) ≥ g2_min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

*Παράδειγμα BOA*ε=0*

επανάληψη 3



Κορυφή εκκίνησης

Κορυφή στόχος

$(s_{goal}, (3,9), (3,9), s_2)$

Open

$(s_1, (1,1), (4,6), s_{start})$

$(s_3, (1,1), (5,6), s_{start})$

0	∞	5	∞	9	g_2^{min}

1^{st}: $(s_{goal}, (3,9), (3,9), s_2)$	*sols*

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

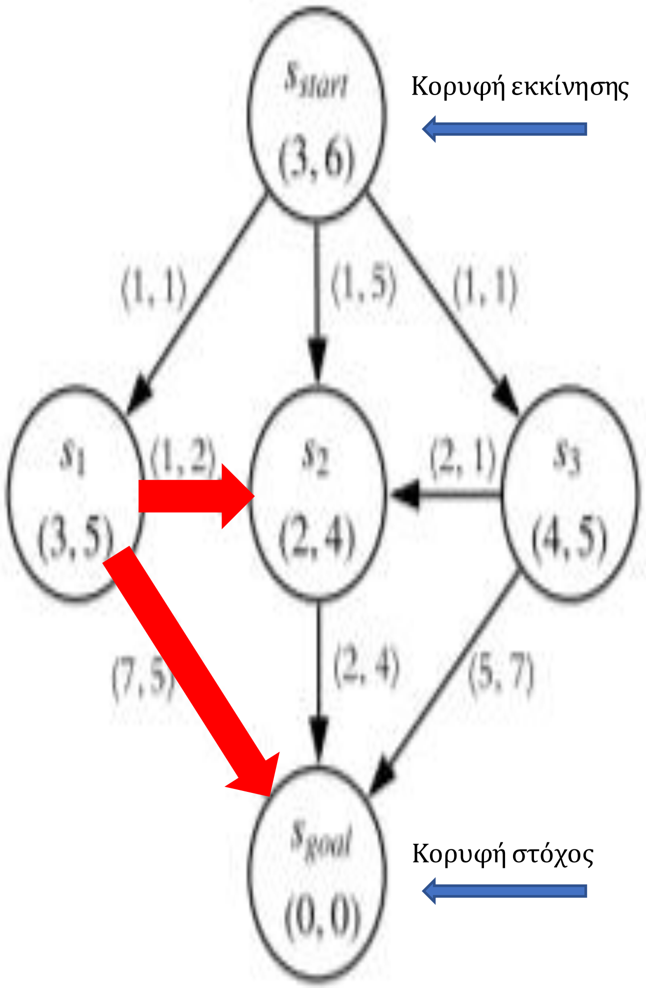
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2min(s(x)) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
        continue
    g2min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2min(t) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
            continue
        Add y to Open
return sols
```

Ψευδοκώδικας BOA*ε

Παράδειγμα BOA*ε=0

επανάληψη 4



Κορυφή εκκίνησης

Κορυφή στόχος

$(s_1, (1, 1), (4, 6), s_{start})$

Open

$(s_2, (2,3), (4,7), s_1)$

$(s_3, (1,1), (5,6), s_{start})$

$(s_{goal}, (8,6), (8,6), s_1)$

0	1	5	∞	9	g_2^{min}

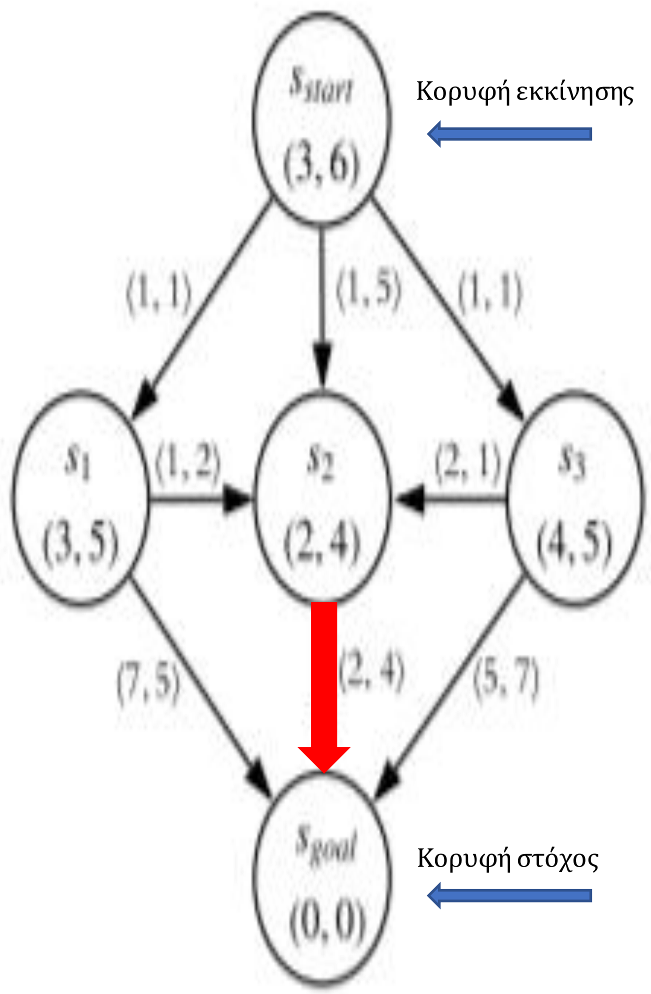
Algorithm 2: Bi-Objective A* (BOA*)

```
Input  : A search problem (S, E, c, s_start, s_goal) and a
         consistent heuristic function h
Output: A cost-unique Pareto-optimal solution set
sols ← ∅
for each s ∈ S do
  └ g2min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
  Remove a node x from Open with the
    lexicographically smallest f-value of all nodes in
    Open
  if g2(x) ≥ g2min(s(x)) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
    └ continue
  g2min(s(x)) ← g2(x)
  if s(x) = s_goal then
    Add x to sols
    └ continue
  for each t ∈ Succ(s(x)) do
    y ← new node with s(y) = t
    g(y) ← g(x) + c(s(x), t)
    parent(y) ← x
    f(y) ← g(y) + h(t)
    if g2(y) ≥ g2min(t) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
      └ continue
    Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

Παράδειγμα BOA*ε=0

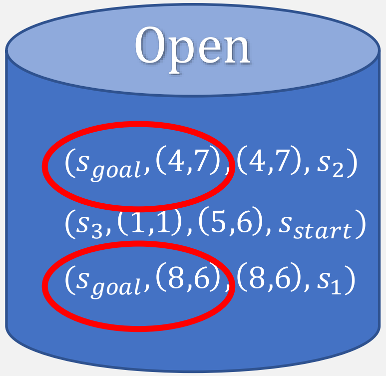
επανάληψη 5



Κορυφή εκκίνησης

Κορυφή στόχος

$(s_2, (2,3), (4,7), s_1)$



0	1	3	∞	9	g_2^{min}

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

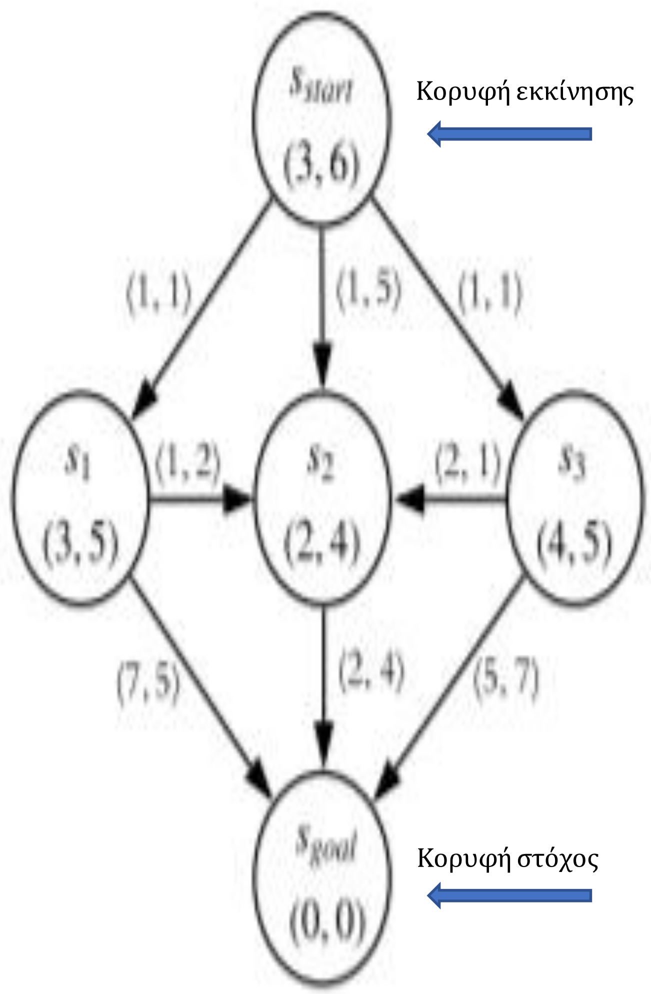
Output: A cost-unique Pareto-optimal solution set

$sols \leftarrow \emptyset$
for each $s \in S$ **do**
  $g_2^{\min}(s) \leftarrow \infty$
$x \leftarrow$ new node with $s(x) = s_{start}$
$\mathbf{g}(x) \leftarrow (0, 0)$
$parent(x) \leftarrow$ null
$\mathbf{f}(x) \leftarrow (h_1(s_{start}), h_2(s_{start}))$
Initialize $Open$ and add x to it
while $Open \neq \emptyset$ **do**
  Remove a node x from $Open$ with the lexicographically smallest f-value of all nodes in $Open$
  if $g_2(x) \geq g_2^{\min}(s(x)) \vee (1+\varepsilon) * f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
    continue
  $g_2^{\min}(s(x)) \leftarrow g_2(x)$
  if $s(x) = s_{goal}$ **then**
    Add x to $sols$
    continue
  for each $t \in \mathrm{Succ}(s(x))$ **do**
    $y \leftarrow$ new node with $s(y) = t$
    $\mathbf{g}(y) \leftarrow \mathbf{g}(x) + \mathbf{c}(s(x), t)$
    $parent(y) \leftarrow x$
    $\mathbf{f}(y) \leftarrow \mathbf{g}(y) + \mathbf{h}(t)$
    if $g_2(y) \geq g_2^{\min}(t) \vee (1+\varepsilon) * f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
      continue
    Add y to $Open$
return $sols$

*Ψευδοκώδικας BOA*ε*

Παράδειγμα BOA*ε=0

επανάληψη 6



Κορυφή εκκίνησης

Κορυφή στόχος

$(\boldsymbol{s_{goal}}, (\mathbf{4}, \mathbf{7}), (\mathbf{4}, \mathbf{7}), \boldsymbol{s_2})$

Open

$(s_3, (1,1), (5,6), s_{start})$

$(s_{goal}, (8,6), (8,6), s_1)$

0	1	3	∞	7	g_2^{min}

1^{st}:	$(s_{goal}, (3,9), (3,9), s_2)$	*sols*
2^{nd}:	$(s_{goal}, (4,7), (4,7), s_2)$	

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

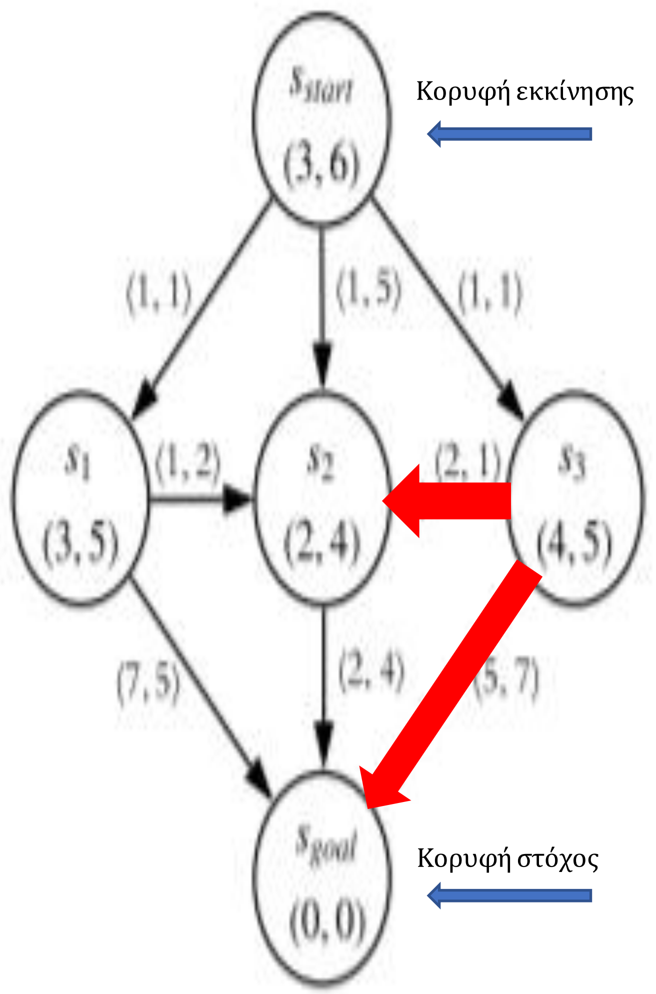
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
     lexicographically smallest f-value of all nodes in
     Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

Παράδειγμα BOA*ε=0

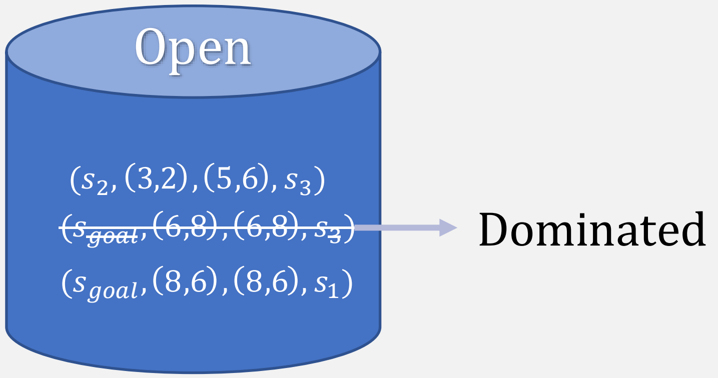
Επανάληψη 7



Κορυφή εκκίνησης

Κορυφή στόχος

$(s_3, (1,1), (5,6), s_{start})$



0	1	3	3	7	g_2^{min}

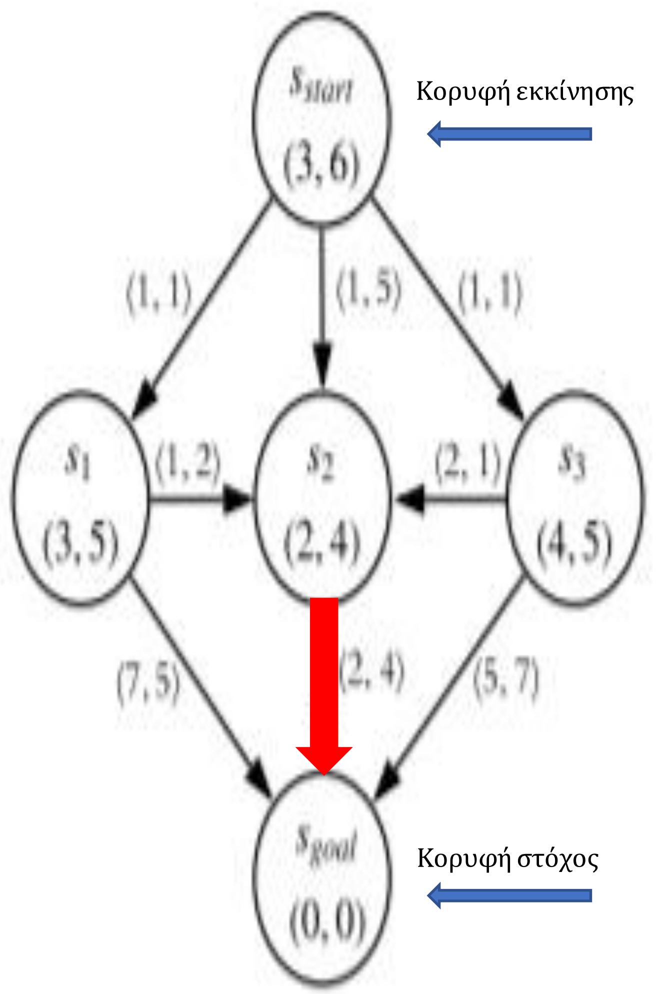
Algorithm 2: Bi-Objective A* (BOA*)

```
Input  : A search problem (S, E, c, s_start, s_goal) and a
         consistent heuristic function h
Output: A cost-unique Pareto-optimal solution set
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
     lexicographically smallest f-value of all nodes in
     Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

*Παράδειγμα BOA*ε=0*

Επανάληψη 8



Κορυφή εκκίνησης

Κορυφή στόχος

$$(s_2, (3, 2), (5, 6), s_3)$$

Open

$(s_{goal}, (5,6), (5,6), s_2)$

$(s_{goal}, (8,6), (8,6), s_1)$

0	1	2	3	7	g_2^{min}

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

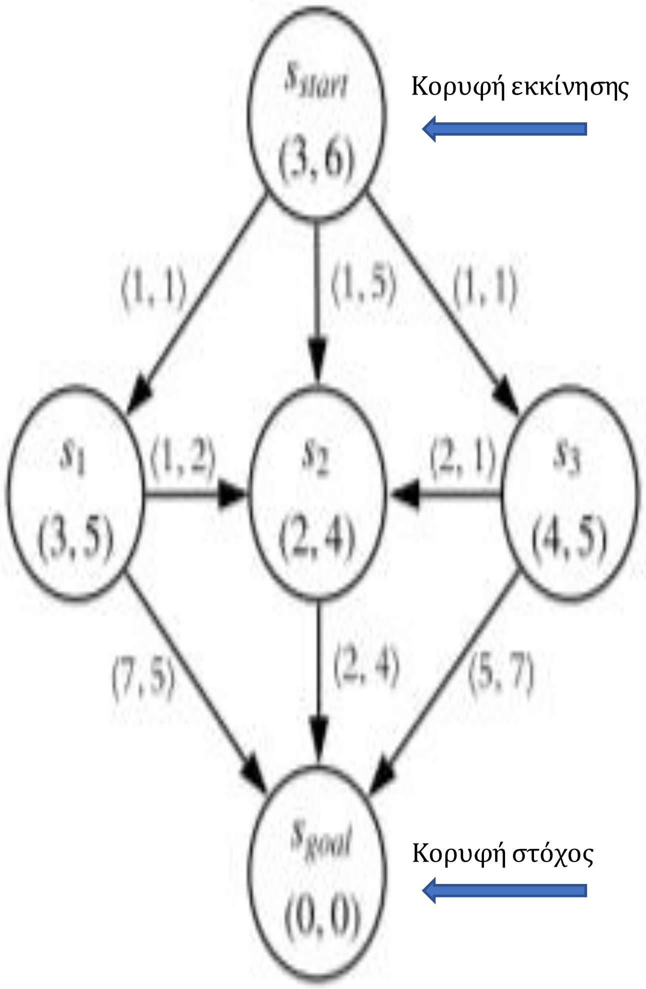
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2min(s(x)) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
        continue
    g2min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2min(t) ∨ (1+ε)*f2(x) ≥ g2min(s_goal) then
            continue
        Add y to Open
return sols
```

*Ψευδοκώδικας BOA*ε*

*Παράδειγμα BOA*ε=0*

Επανάληψη 9



Κορυφή εκκίνησης

Κορυφή στόχος

$(\boldsymbol{s_{goal}}, (\mathbf{5}, \mathbf{6}), (\mathbf{5}, \mathbf{6}), \boldsymbol{s_2})$

Open

$(s_{goal}, (8,6), (8,6), s_1)$

0	1	2	3	6	g_2^{min}

sols	
1^{st}:	$(s_{goal}, (3,9), (3,9), s_2)$
2^{nd}:	$(s_{goal}, (4,7), (4,7), s_2)$
3^{rd}:	$(s_{goal}, (5,6), (5,6), s_2)$

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$

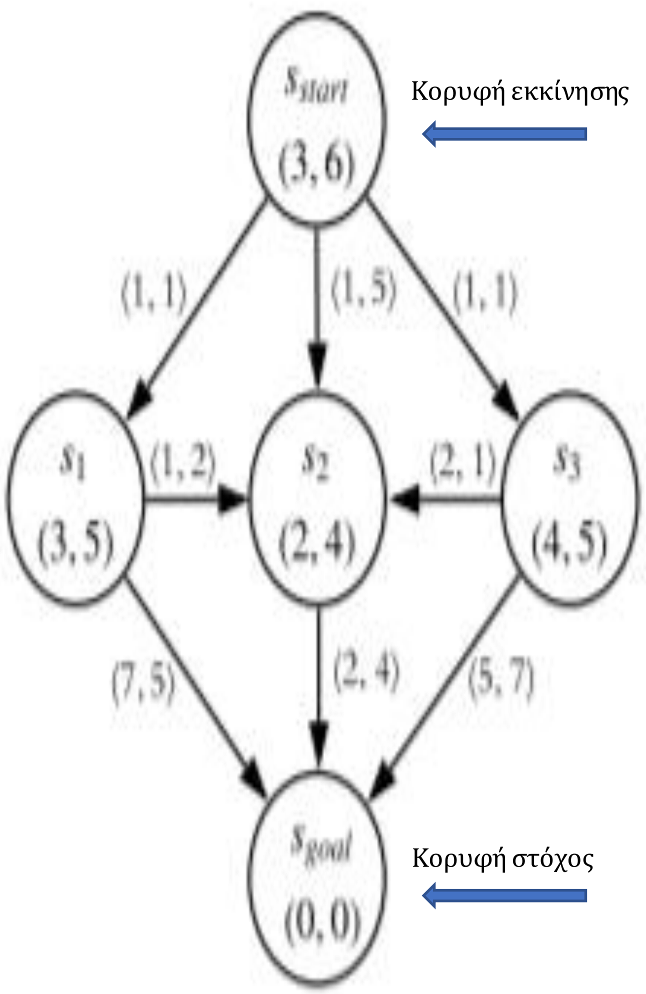
Output: A cost-unique Pareto-optimal solution set

```
sols ← ∅
for each s ∈ S do
    g2^min(s) ← ∞
x ← new node with s(x) = s_start
g(x) ← (0, 0)
parent(x) ← null
f(x) ← (h1(s_start), h2(s_start))
Initialize Open and add x to it
while Open ≠ ∅ do
    Remove a node x from Open with the
      lexicographically smallest f-value of all nodes in
      Open
    if g2(x) ≥ g2^min(s(x)) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
        continue
    g2^min(s(x)) ← g2(x)
    if s(x) = s_goal then
        Add x to sols
        continue
    for each t ∈ Succ(s(x)) do
        y ← new node with s(y) = t
        g(y) ← g(x) + c(s(x), t)
        parent(y) ← x
        f(y) ← g(y) + h(t)
        if g2(y) ≥ g2^min(t) ∨ (1+ε)*f2(x) ≥ g2^min(s_goal) then
            continue
        Add y to Open
return sols
```

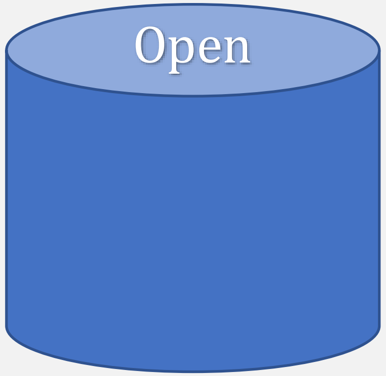
Ψευδοκώδικας BOA*ε

Παράδειγμα BOA*ε=0

Επανάληψη 10



$(s_{goal}, (8, 6), (8, 6), s_1)$ **-> Dominated**



Return

Pareto -optimal frontier

1^{st}: $(s_{goal}, (3,9), (3,9), s_2)$
2^{nd}: $(s_{goal}, (4,7), (4,7), s_2)$
3^{rd}: $(s_{goal}, (5,6), (5,6), s_2)$

Algorithm 2: Bi-Objective A* (BOA*)

Input : A search problem $(S, E, \mathbf{c}, s_{start}, s_{goal})$ and a consistent heuristic function $\mathbf{h}$
Output: A cost-unique Pareto-optimal solution set

$sols \leftarrow \emptyset$
for each $s \in S$ **do**
  $g_2^{\min}(s) \leftarrow \infty$
$x \leftarrow$ new node with $s(x) = s_{start}$
$\mathbf{g}(x) \leftarrow (0, 0)$
$parent(x) \leftarrow$ null
$\mathbf{f}(x) \leftarrow (h_1(s_{start}), h_2(s_{start}))$
Initialize $Open$ and add x to it
while $Open \neq \emptyset$ **do**
  Remove a node x from $Open$ with the lexicographically smallest f-value of all nodes in $Open$
  if $g_2(x) \geq g_2^{\min}(s(x)) \vee (1+\varepsilon)*f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
    continue
  $g_2^{\min}(s(x)) \leftarrow g_2(x)$
  if $s(x) = s_{goal}$ **then**
    Add x to $sols$
    continue
  for each $t \in \mathrm{Succ}(s(x))$ **do**
    $y \leftarrow$ new node with $s(y) = t$
    $\mathbf{g}(y) \leftarrow \mathbf{g}(x) + \mathbf{c}(s(x), t)$
    $parent(y) \leftarrow x$
    $\mathbf{f}(y) \leftarrow \mathbf{g}(y) + \mathbf{h}(t)$
    if $g_2(y) \geq g_2^{\min}(t) \vee (1+\varepsilon)*f_2(x) \geq g_2^{\min}(s_{goal})$ **then**
      continue
    Add y to $Open$
return $sols$

*Ψευδοκώδικας BOA*ε*

Παράδειγμα PPA ε=0*

κόμβος *x* του PPA*

Αντιστοιχεί σε ζεύγος μονοπατιών

tl: $s(x), (g_1, g_2), (f_1, f_2), p(x)$
br: $s(x), (g_1, g_2), (f_1, f_2), p(x)$

Κόμβος BOA* -> μία διαδρομή μονοπάτι
Κόμβος PPA* -> ζεύγος διαδρομών μονοπατιών

2 επιπλέον λειτουργίες

Συγχώνευση

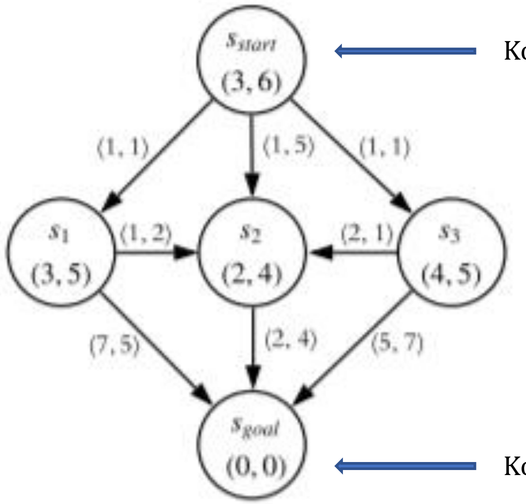
- $if\ c_{1\backslash 2}\left(\pi_v^{tl\backslash br}\right) \leq c_{1\backslash 2}\left(\pi_v'^{\,tl\backslash br}\right) \rightarrow \pi_{merged}^{tl\backslash br} = \pi^{tl\backslash br}$
- $else \rightarrow \pi_{merged}^{tl\backslash \mathrm{br}} = \pi'^{tl\backslash \mathrm{br}}$

Οριοθέτηση

- $\varepsilon_1 \geq (c_1(\pi_{merged}^{br}) - \mathrm{c}_1(\pi_{\mathrm{merged}}^{\mathrm{tl}}))/\mathrm{c}_1(\pi_{\mathrm{merged}}^{\mathrm{tl}})$
- $\varepsilon_2 \geq (c_2(\pi_{merged}^{tl}) - \mathrm{c}_2(\pi_{\mathrm{merged}}^{\mathrm{br}}))/\mathrm{c}_2(\pi_{\mathrm{merged}}^{\mathrm{br}})$

Αρχικοποίηση

Παράδειγμα PPA ε=0



Κορυφή εκκίνησης

Κορυφή στόχος

λίστα sols_pp	→	*Πιθανές! λύσεις*
λίστα g_2^{min}		
Αρχικός κόμβος	→	***tl***: $(s_{start}, (0,0), (3,6), null)$ ***br***: $(s_{start}, (0,0), (3,6), null)$
OpenList	→	(f_1^{tl}, f_2^{br}) *λεξικογραφική ταξινόμηση*

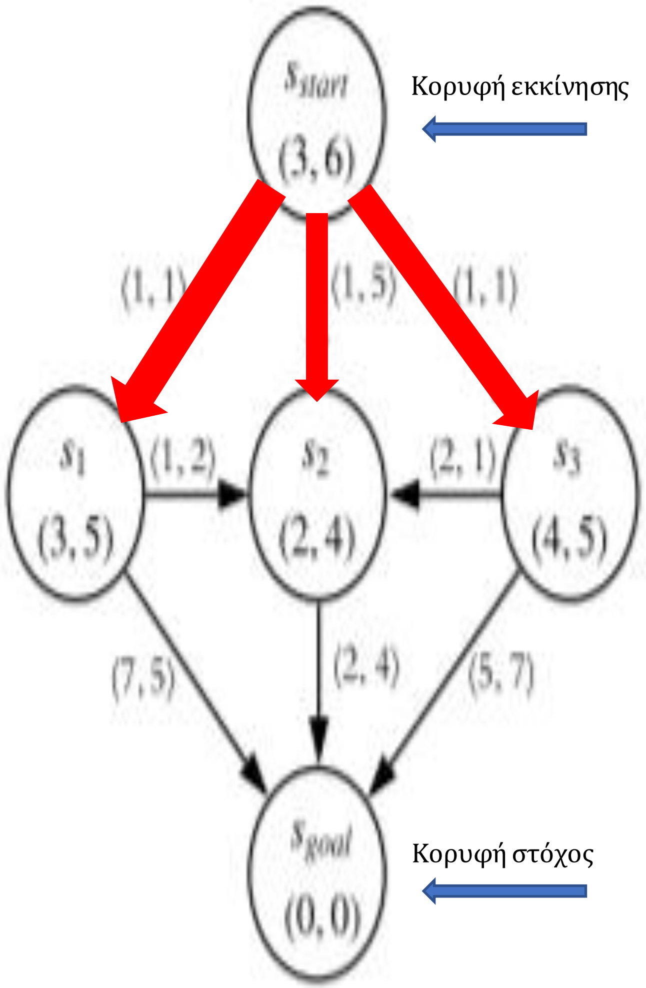
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

solutions_pp ← ∅ ▷ path pairs
OPEN ← new path pair (v_{start}, v_{start})
while OPEN ≠ ∅ **do**
  (π_u^{tl}, π_u^{br}) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^{tl}, π_u^{br}) **then**
    continue
  if $u = v_{goal}$ **then** ▷ reached goal
    merge_to_solutions_PP-A*$(\pi_u^{tl}, \pi_u^{br}$, solutions_pp)
    continue
  for $e = (u, v) \in$ neighbors$(s(n), G)$ **do**
    π_v^{tl}, π_v^{br} ← extend_PP-A*$((\pi_u^{tl}, \pi_u^{br}), e)$
    if is_dominated_PP-A*(π_v^{tl}, π_v^{br}) **then**
      continue
    insert_PP-A*$((\pi_v^{tl}, \pi_v^{br})$, OPEN)
solutions ← ∅
for $(\pi_{v_{goal}}^{tl}, \pi_{v_{goal}}^{br}) \in$ solutions_pp **do**
  solutions ← solutions $\cup \{\pi_{v_{goal}}^{tl}\}$
return solutions

Ψευδοκώδικας PPA*

Παράδειγμα PPA*ε=0



Κορυφή εκκίνησης

Κορυφή στόχος

επανάληψη 1

$(s_{start}, (0,0), (3,6), null)$
$(s_{start}, (0,0), (3,6), null)$

Open

$tl: (s_{start}, (0,0), (3,6), null)$
$br: (s_{start}, (0,0), (3,6), null)$

$tl: (s_2, (1,5), (3,9), s_{start})$
$br: (s_2, (1,5), (3,9), s_{start})$

$tl: (s_1, (1,1), (4,6), s_{start})$
$br: (s_1, (1,1), (4,6), s_{start})$

$tl: (s_3, (1,1), (5,6), s_{start})$
$br: (s_3, (1,1), (5,6), s_{start})$

0	∞	∞	∞	∞

g_2^{min}

Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

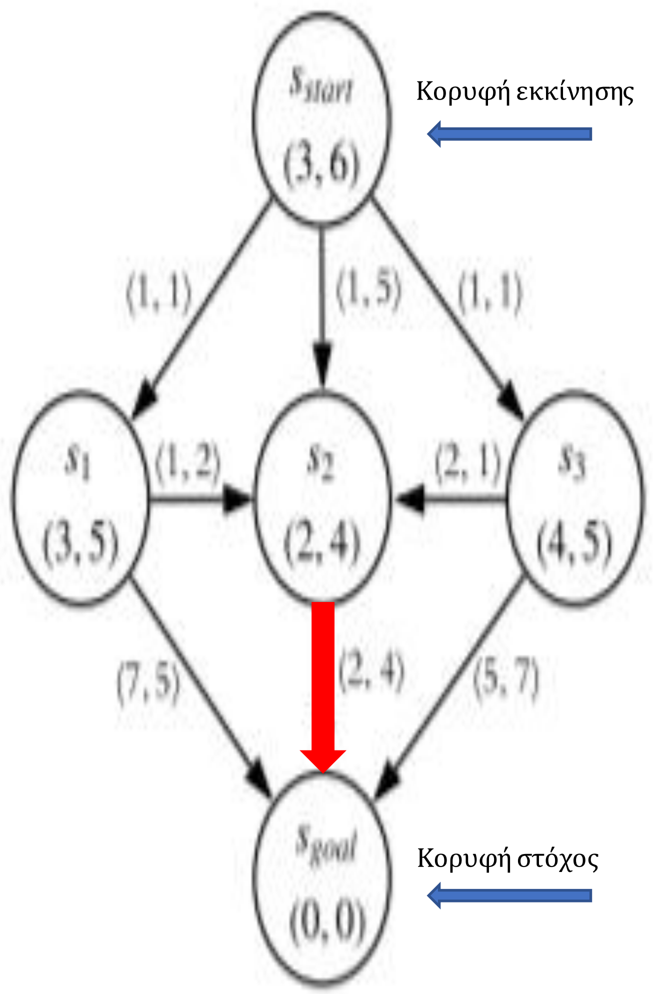
```
solutions_pp ← ∅                                        ▷ path pairs
OPEN ← new path pair (v_start, v_start)
while OPEN ≠ ∅ do
  (π_u^tl, π_u^br) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^tl, π_u^br) then
    continue
  if u = v_goal then                                    ▷ reached goal
    merge_to_solutions_PP-A*(π_u^tl, π_u^br, solutions_pp)
    continue
  for e = (u, v) ∈ neighbors(s(n), G) do
    π_v^tl, π_v^br ← extend_PP-A*((π_u^tl, π_u^br), e)
    if is_dominated_PP-A*(π_v^tl, π_v^br) then
      continue
    insert_PP-A*((π_v^tl, π_v^br), OPEN)
solutions ← ∅
for (π_{v_goal}^tl, π_{v_goal}^br) ∈ solutions_pp do
  solutions ← solutions ∪ {π_{v_goal}^tl}
return solutions
```

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0

επανάληψη 2

$(s_2, (1,5), (3,9), s_{start})$
$(s_2, (1,5), (3,9), s_{start})$



Open

$tl: (s_{goal}, (3,9), (3,9), s_2)$
$br: (s_{goal}, (3,9), (3,9), s_2)$

$tl: (s_1, (1,1), (4,6), s_{start})$
$(br: (s_1, (1,1), (4,6), s_{start})$

$tl: (s_3, (1,1), (5,6), s_{start})$
$br: (s_3, (1,1), (5,6), s_{start})$

0	∞	5	∞	∞	g_2^{min}

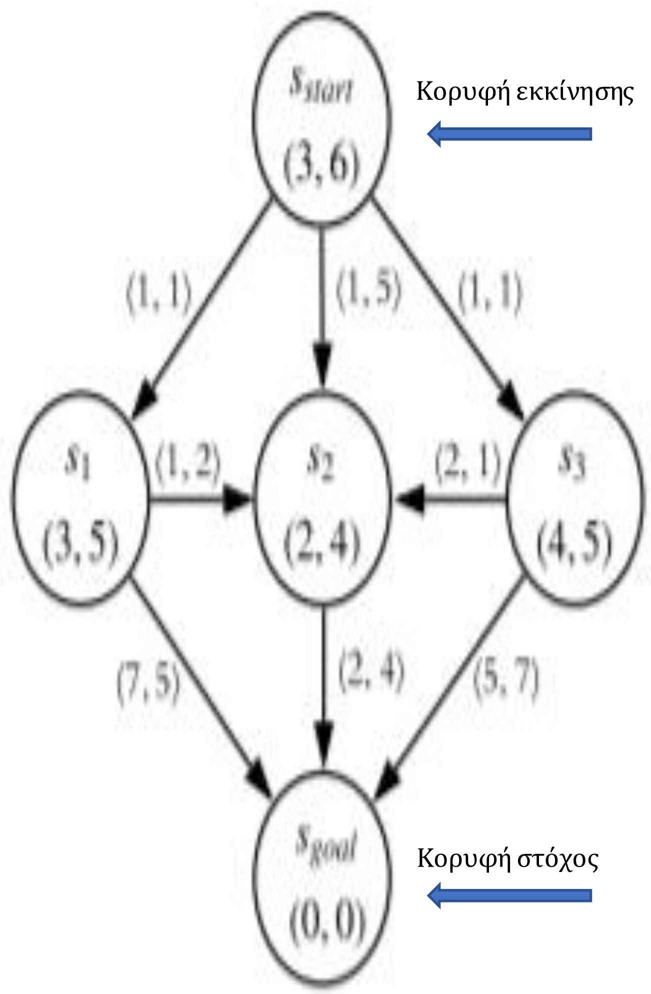
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

```
solutions_pp ← ∅                                         ▷ path pairs
OPEN ← new path pair (v_start, v_start)
while OPEN ≠ ∅ do
  (π_u^tl, π_u^br) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^tl, π_u^br) then
    continue
  if u = v_goal then                                     ▷ reached goal
    merge_to_solutions_PP-A*(π_u^tl, π_u^br, solutions_pp)
    continue
  for e = (u, v) ∈ neighbors(s(n), G) do
    π_v^tl, π_v^br ← extend_PP-A*((π_u^tl, π_u^br), e)
    if is_dominated_PP-A*(π_v^tl, π_v^br) then
      continue
    insert_PP-A*((π_v^tl, π_v^br), OPEN)
solutions ← ∅
for (π_{v_goal}^tl, π_{v_goal}^br) ∈ solutions_pp do
  solutions ← solutions ∪ {π_{v_goal}^tl}
return solutions
```

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



επανάληψη 3

$(s_{goal}, (3,9), (3,9), s_2)$
$(s_{goal}, (3,9), (3,9), s_2)$

Open

$tl: (s_1, (1,1), (4,6), s_{start})$
$(br: (s_1, (1,1), (4,6), s_{start})$

$tl: (s_3, (1,1), (5,6), s_{start})$
$br: (s_3, (1,1), (5,6), s_{start})$

0	∞	5	∞	9	g_2^{min}

Sols_pp

1^{st}: $(s_{goal}, (3,9), (3,9), s_2)$, $(s_{goal}, (3,9), (3,9), s_2)$

Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

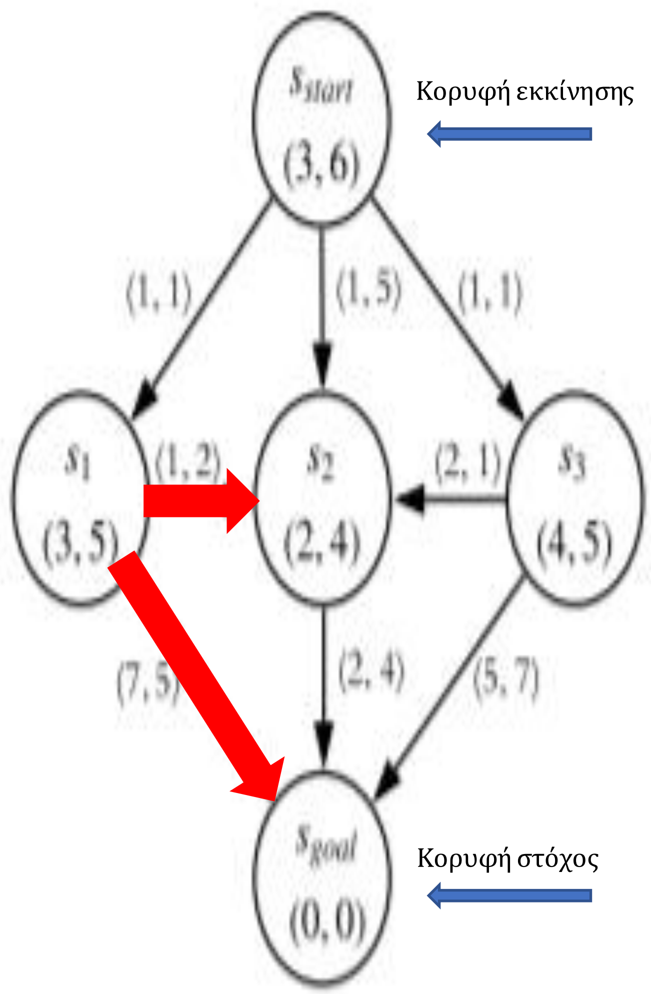
solutions_pp ← ∅ ▷ path pairs
OPEN ← new path pair (v_{start}, v_{start})
while OPEN ≠ ∅ **do**
  (π_u^{tl}, π_u^{br}) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^{tl}, π_u^{br}) **then**
    continue
  if $u = v_{goal}$ **then** ▷ reached goal
    merge_to_solutions_PP-A*$(\pi_u^{tl}, \pi_u^{br}$, solutions_pp)
    continue
  for $e = (u, v)$ ∈ neighbors$(s(n), G)$ **do**
    π_v^{tl}, π_v^{br} ← extend_PP-A*$((\pi_u^{tl}, \pi_u^{br}), e)$
    if is_dominated_PP-A*(π_v^{tl}, π_v^{br}) **then**
      continue
    insert_PP-A*$((\pi_v^{tl}, \pi_v^{br})$, OPEN)
solutions ← ∅
for $(\pi_{v_{goal}}^{tl}, \pi_{v_{goal}}^{br})$ ∈ solutions_pp **do**
  solutions ← solutions ∪ $\{\pi_{v_{goal}}^{tl}\}$
return solutions

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0

επανάληψη 4

$(s_1, (1,1), (4,6), s_{start})$
$(s_1, (1,1), (4,6), s_{start})$



Κορυφή εκκίνησης

Κορυφή στόχος

Open

$tl: (s_2, (2,3), (4,7), s_1)$
$br: (s_2, (2,3), (4,7), s_1)$

$tl: (s_3, (1,1), (5,6), s_{start})$
$br: (s_3, (1,1), (5,6), s_{start})$

$tl: (s_{goal}, (8,6), (8,6), s_1)$
$br\,(s_{goal}, (8,6), (8,6), s_1)$

0	1	5	∞	9	g_2^{min}

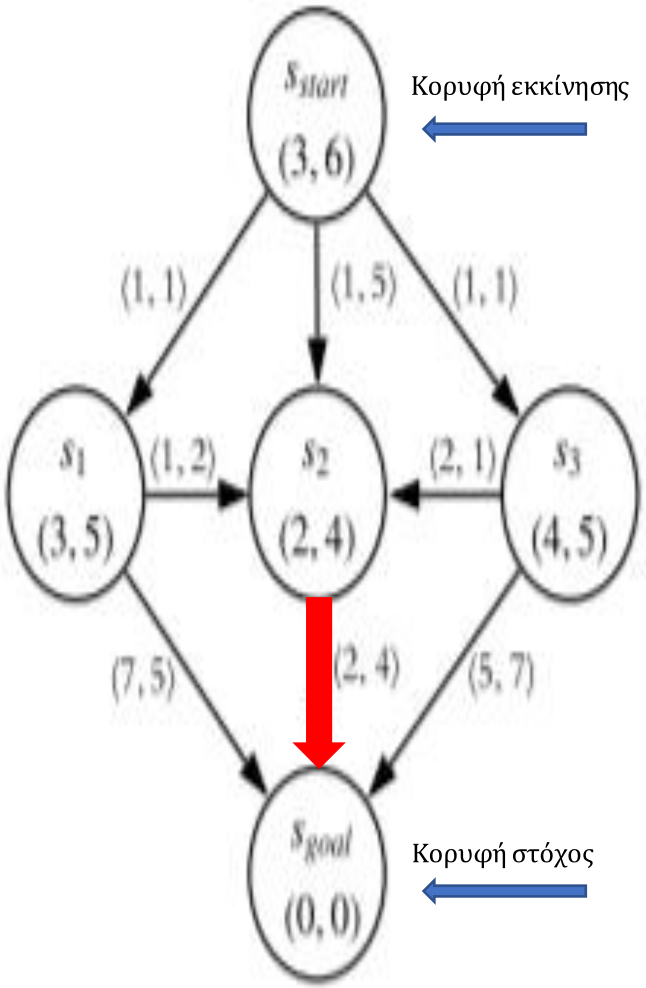
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

solutions_pp $\leftarrow \emptyset$ ▷ path pairs
OPEN $\leftarrow$ new path pair (v_{start}, v_{start})
while OPEN $\neq \emptyset$ **do**
  $(\pi_u^{tl}, \pi_u^{br}) \leftarrow$ OPEN.extract_min()
  if is_dominated_PP-A*(π_u^{tl}, π_u^{br}) **then**
    continue
  if $u = v_{goal}$ **then** ▷ reached goal
    merge_to_solutions_PP-A*$(\pi_u^{tl}, \pi_u^{br}$, solutions_pp)
    continue
  for $e = (u, v) \in$ neighbors$(s(n), G)$ **do**
    $\pi_v^{tl}, \pi_v^{br} \leftarrow$ extend_PP-A*$((\pi_u^{tl}, \pi_u^{br}), e)$
    if is_dominated_PP-A*(π_v^{tl}, π_v^{br}) **then**
      continue
    insert_PP-A*$((\pi_v^{tl}, \pi_v^{br})$, OPEN)
solutions $\leftarrow \emptyset$
for $(\pi_{v_{goal}}^{tl}, \pi_{v_{goal}}^{br}) \in$ solutions_pp **do**
  solutions $\leftarrow$ solutions $\cup \{\pi_{v_{goal}}^{tl}\}$
return solutions

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



Κορυφή εκκίνησης

Κορυφή στόχος

επανάληψη 5

$(s_2, (2, 3), (4, 7), s_1)$

$(s_2, (2, 3), (4, 7), s_1)$

Open

$tl: (s_{goal}, (4,7), (4,7), s_2)$
$br: (s_{goal}, (4,7), (4,7), s_2)$

$tl: (s_3, (1,1), (5,6), s_{start})$
$br: (s_3, (1,1), (5,6), s_{start})$

$tl: (s_{goal}, (8,6), (8,6), s_1)$
$br\,(s_{goal}, (8,6), (8,6), s_1)$

Merging merging

$tl: (s_{goal}, (4,7), (4,7), s_2)$
$br: (s_{goal}, (4,7), (4,7), s_2)$

$tl: (s_{goal}, (8, 6), (8, 6), s_1)$
$br\,(s_{goal}, (8, 6), (8, 6), s_1)$

Not bounded

0	1	3	∞	9

g_2^{min}

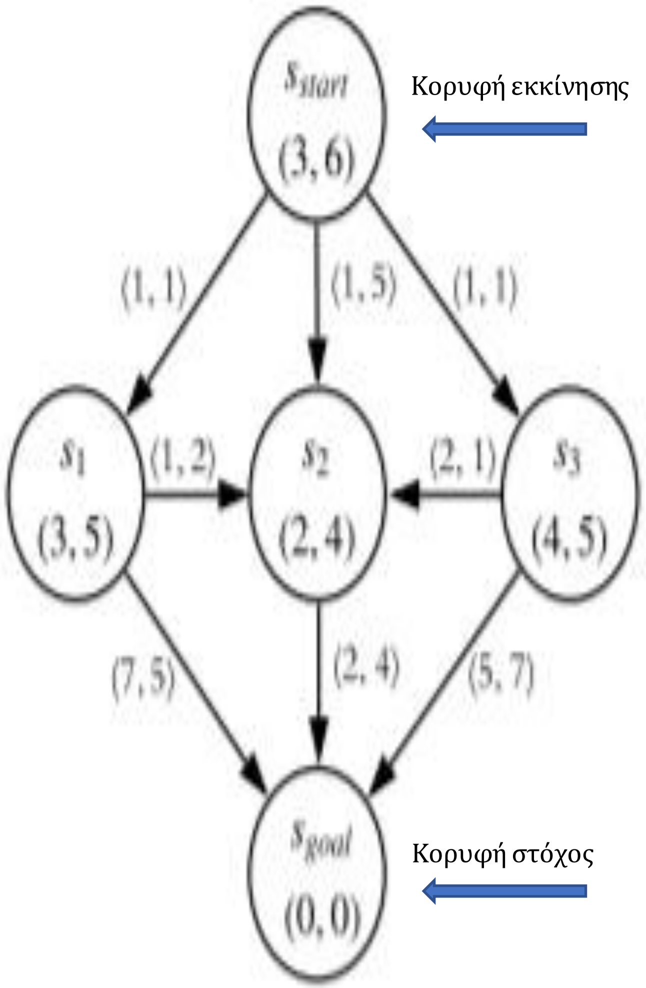
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

```
solutions_pp ← ∅                                    ▷ path pairs
OPEN ← new path pair (v_start, v_start)
while OPEN ≠ ∅ do
  (π_u^tl, π_u^br) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^tl, π_u^br) then
    continue
  if u = v_goal then                                ▷ reached goal
    merge_to_solutions_PP-A*(π_u^tl, π_u^br, solutions_pp)
    continue
  for e = (u, v) ∈ neighbors(s(n), G) do
    π_v^tl, π_v^br ← extend_PP-A*((π_u^tl, π_u^br), e)
    if is_dominated_PP-A*(π_v^tl, π_v^br) then
      continue
    insert_PP-A*((π_v^tl, π_v^br), OPEN)
solutions ← ∅
for (π_{v_goal}^tl, π_{v_goal}^br) ∈ solutions_pp do
  solutions ← solutions ∪ {π_{v_goal}^tl}
return solutions
```

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



επανάληψη 6

$(s_{goal}, (4,7), (4,7), s_2)$
$(s_{goal}, (4,7), (4,7), s_2)$

Open

tl: $(s_3, (1,1), (5,6), s_{start})$
br: $(s_3, (1,1), (5,6), s_{start})$

tl: $(s_{goal}, (8,6), (8,6), s_1)$
br $(s_{goal}, (8,6), (8,6), s_1)$

merging

~~Merged node~~
tl: $(s_{goal}, (3,9), (3,9), s_2)$
br: $(s_{goal}, (4,7), (4,7), s_2)$

tl: $(s_{goal}, (4,7), (4,7), s_2)$
br: $(s_{goal}, (4,7), (4,7), s_2)$

Not bounded

0	1	3	∞	7	g_2^{min}

Sols_pp

1^{st}: $(s_{goal}, (3,9), (3,9), s_2), (s_{goal}, (3,9), (3,9), s_2)$
2^{nd}: $(s_{goal}, (4,7), (4,7), s_2), (s_{goal}, (4,7), (4,7), s_2)$

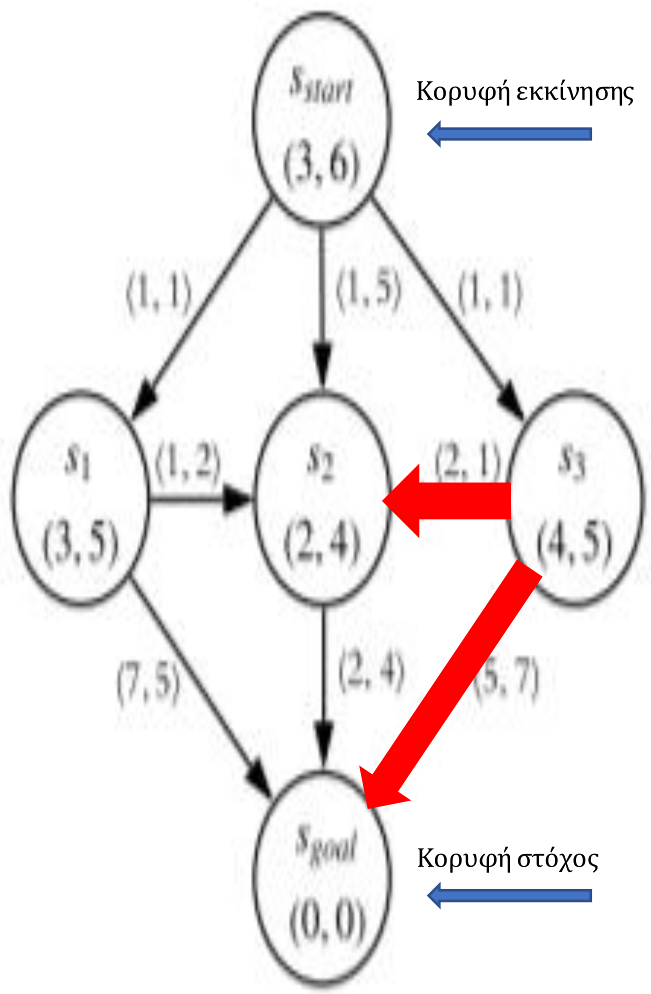
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

solutions_pp ← $\emptyset$ ▷ path pairs
OPEN ← new path pair (v_{start}, v_{start})
while OPEN $\neq \emptyset$ **do**
  (π_u^{tl}, π_u^{br}) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^{tl}, π_u^{br}) **then**
    continue
  if $u = v_{goal}$ **then** ▷ reached goal
    merge_to_solutions_PP-A*$(\pi_u^{tl}, \pi_u^{br}$, solutions_pp)
    continue
  for $e = (u, v) \in$ neighbors$(s(n), G)$ **do**
    π_v^{tl}, π_v^{br} ← extend_PP-A*$((\pi_u^{tl}, \pi_u^{br}), e)$
    if is_dominated_PP-A*(π_v^{tl}, π_v^{br}) **then**
      continue
    insert_PP-A*$((\pi_v^{tl}, \pi_v^{br})$, OPEN)
solutions ← $\emptyset$
for $(\pi_{v_{goal}}^{tl}, \pi_{v_{goal}}^{br}) \in$ solutions_pp **do**
  solutions ← solutions $\cup \{\pi_{v_{goal}}^{tl}\}$
return solutions

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



Επανάληψη 7

$(\boldsymbol{s_3}, (\mathbf{1}, \mathbf{1}), (\mathbf{5}, \mathbf{6}), \boldsymbol{s_{start}})$
$(\boldsymbol{s_3}, (\mathbf{1}, \mathbf{1}), (\mathbf{5}, \mathbf{6}), \boldsymbol{s_{start}})$

Open

$tl: (s_2, (3,2), (5,6), s_3)$
$br: (s_2, (3,2), (5,6), s_3)$

$\cancel{tl: (s_{goal}, (6,8), (6,8), s_3)}$
$\cancel{br: (s_{goal}, (6,8), (6,8), s_3)}$

$tl: (s_{goal}, (8,6), (8,6), s_1)$
$br\,(s_{goal}, (8,6), (8,6), s_1)$

0	1	3	3	7	g_2^{min}

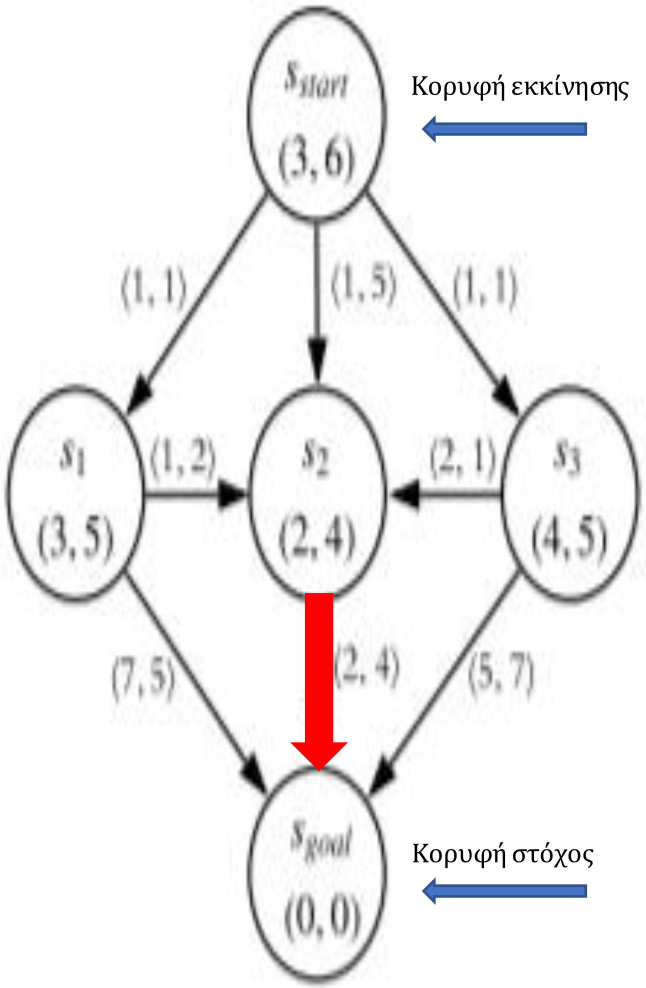
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

```
solutions_pp ← ∅                                   ▷ path pairs
OPEN ← new path pair (v_start, v_start)
while OPEN ≠ ∅ do
  (π_u^tl, π_u^br) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^tl, π_u^br) then
    continue
  if u = v_goal then                               ▷ reached goal
    merge_to_solutions_PP-A*(π_u^tl, π_u^br, solutions_pp)
    continue
  for e = (u, v) ∈ neighbors(s(n), G) do
    π_v^tl, π_v^br ← extend_PP-A*((π_u^tl, π_u^br), e)
    if is_dominated_PP-A*(π_v^tl, π_v^br) then
      continue
    insert_PP-A*((π_v^tl, π_v^br), OPEN)
solutions ← ∅
for (π_{v_goal}^tl, π_{v_goal}^br) ∈ solutions_pp do
  solutions ← solutions ∪ {π_{v_goal}^tl}
return solutions
```

*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



Κορυφή εκκίνησης

Κορυφή στόχος

Επανάληψη 8

$(s_2, (3, 2), (5, 6), s_3)$

$(s_2, (3, 2), (5, 6), s_3)$

Open

$tl: (s_{goal}, (5,6), (5,6), s_2)$
$br: (s_{goal}, (5,6), (5,6), s_2)$

$tl: (s_{goal}, (8,6), (8,6), s_1)$
$br(s_{goal}, (8,6), (8,6), s_1)$

Merging

$tl: (s_{goal}, (5,6), (5,6), s_2)$
$br: (s_{goal}, (5,6), (5,6), s_2)$

$tl: (s_{goal}, (8,6), (8,6), s_1)$
$br(s_{goal}, (8,6), (8,6), s_1)$

Bounded

0	1	2	3	7

g_2^{min}

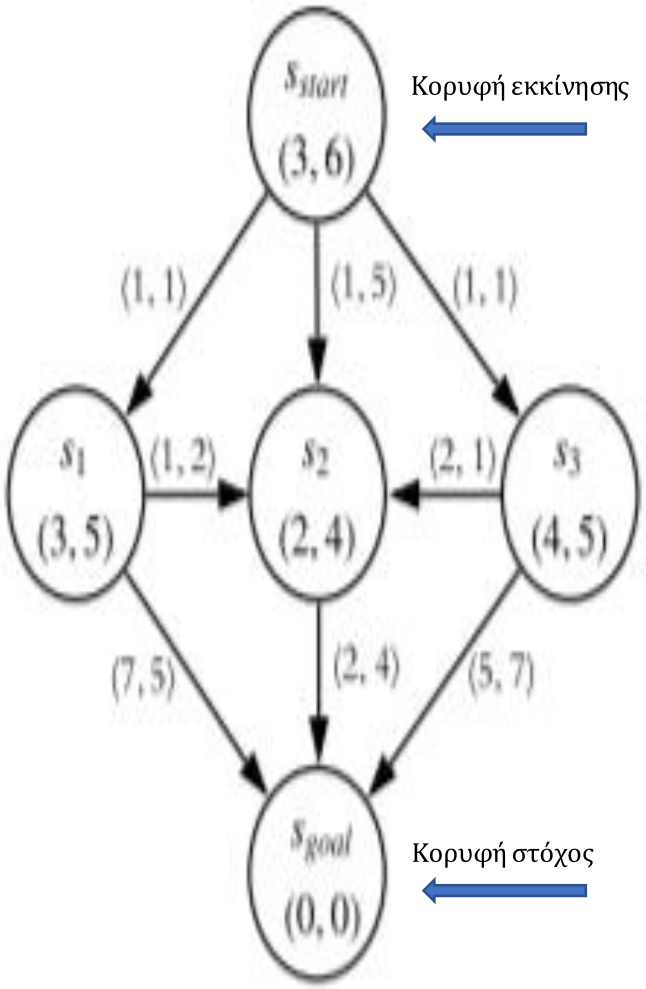
Algorithm 2 PP-A*

Input: $(G = (V, E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

```
solutions_pp ← ∅                                         ▷ path pairs
OPEN ← new path pair (v_start, v_start)
while OPEN ≠ ∅ do
  (π_u^tl, π_u^br) ← OPEN.extract_min()
  if is_dominated_PP-A*(π_u^tl, π_u^br) then
    continue
  if u = v_goal then                                     ▷ reached goal
    merge_to_solutions_PP-A*(π_u^tl, π_u^br, solutions_pp)
    continue
  for e = (u, v) ∈ neighbors(s(n), G) do
    π_v^tl, π_v^br ← extend_PP-A*((π_u^tl, π_u^br), e)
    if is_dominated_PP-A*(π_v^tl, π_v^br) then
      continue
    insert_PP-A*((π_v^tl, π_v^br), OPEN)
solutions ← ∅
for (π_{v_goal}^tl, π_{v_goal}^br) ∈ solutions_pp do
  solutions ← solutions ∪ {π_{v_goal}^tl}
return solutions
```

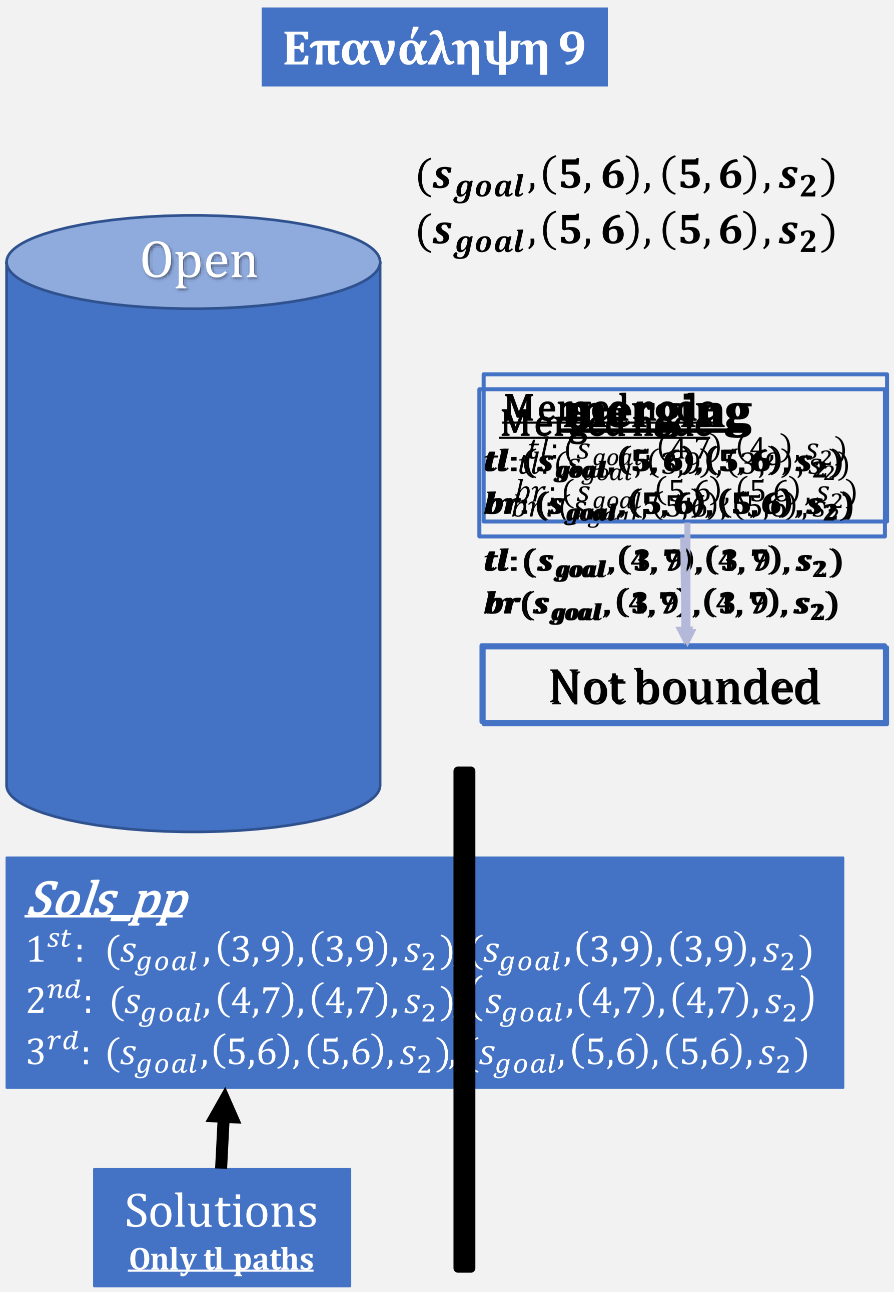
*Ψευδοκώδικας PPA*ε*

Παράδειγμα PPA*ε=0



Κορυφή εκκίνησης

Κορυφή στόχος



Algorithm 2 PP-A*

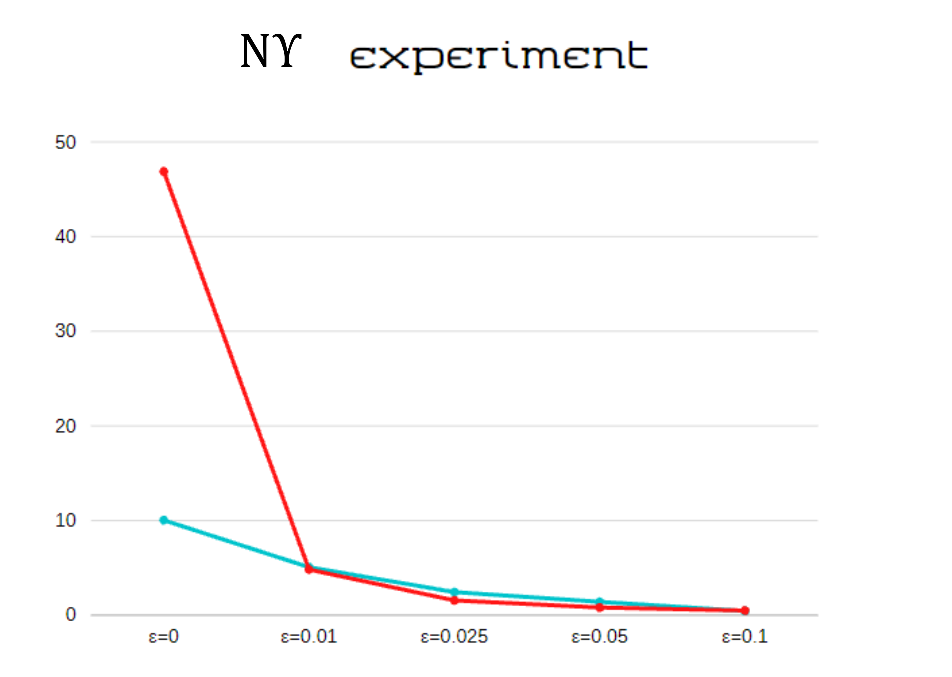
Input: $(G=(V,E), v_{start}, v_{goal}, c_1, c_2, h_1, h_2, \varepsilon_1, \varepsilon_2)$

solutions_pp $\leftarrow \emptyset$ ▷ path pairs
OPEN $\leftarrow$ new path pair (v_{start}, v_{start})
while OPEN $\neq \emptyset$ **do**
  $(\pi_u^{tl}, \pi_u^{br}) \leftarrow$ OPEN.extract_min()
  if is_dominated_PP-A*(π_u^{tl}, π_u^{br}) **then**
    continue
  if $u = v_{goal}$ **then** ▷ reached goal
    merge_to_solutions_PP-A*$(\pi_u^{tl}, \pi_u^{br}$, solutions_pp)
    continue
  for $e = (u, v) \in$ neighbors$(s(n), G)$ **do**
    $\pi_v^{tl}, \pi_v^{br} \leftarrow$ extend_PP-A*$((\pi_u^{tl}, \pi_u^{br}), e)$
    if is_dominated_PP-A*(π_v^{tl}, π_v^{br}) **then**
      continue
    insert_PP-A*$((\pi_v^{tl}, \pi_v^{br})$, OPEN)
solutions $\leftarrow \emptyset$
for $(\pi_{v_{goal}}^{tl}, \pi_{v_{goal}}^{br}) \in$ solutions_pp **do**
  solutions $\leftarrow$ solutions $\cup \{\pi_{v_{goal}}^{tl}\}$
return solutions

*Ψευδοκώδικας PPA*ε*

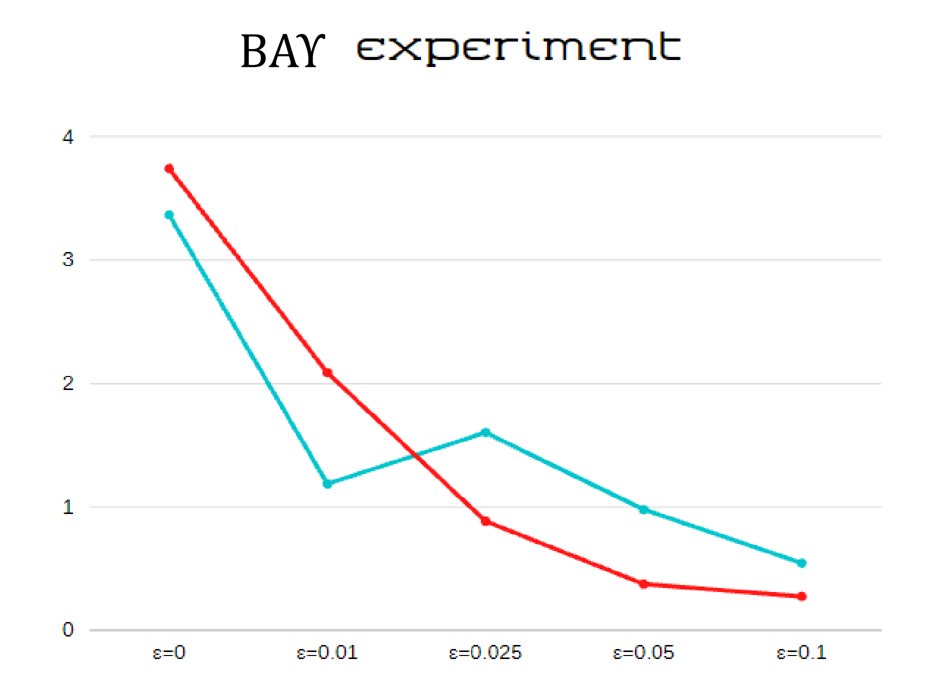
Πείραμα
ΝΕΑ ΥΟΡΚΗ

ε	BOA*$_ε$ num$_{avg}$ sols	PPA* num$_{avg}$ sols	BOA*$_ε$ time$_{avg}$	PPA* time$_{avg}$	BOA*$_ε$ num$_{med}$ sols	PPA* num$_{med}$ sols	BOA*$_ε$ time$_{med}$	PPA* time$_{med}$
ε=0	246,667	246,667	10,0375	46,8853	209	209	11,335	38,317
ε=0.01	29,5	27,1667	5,0695	4,8239	29	27	2,9767	1,9325
ε=0.025	14,5	13,6667	2,4415	1,5648	14	14	0,5482	0,7506
ε=0.05	8,5	7,8333	1,4129	0,8116	9	5	0,4491	0,3635
ε=0.1	4,8334	4,8334	0,4424	0,4920	5	3	0,0196	0,0451

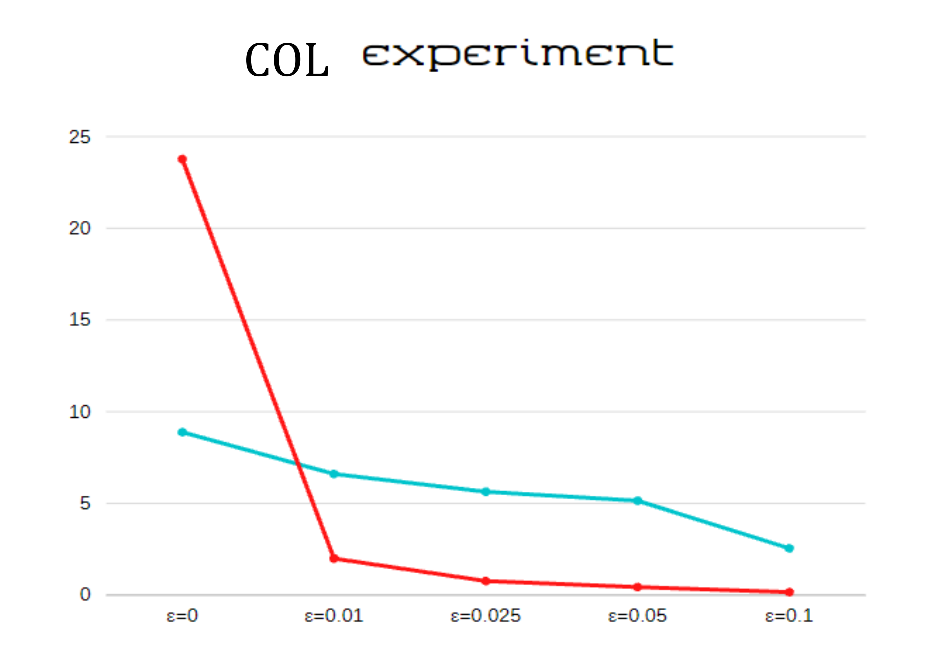


Πείραμα ΣΑΝ ΦΡΑΝΣΙΣΚΟ

ε	BOA*$_ε$ num$_{avg}$ sols	PPA* num$_{avg}$ sols	BOA*$_ε$ time$_{avg}$	PPA* time$_{avg}$	BOA*$_ε$ num$_{med}$ sols	PPA* num$_{med}$ sols	BOA*$_ε$ time$_{med}$	PPA* time$_{med}$
ε=0	230,625	230,625	3,365	3,7402	228	228	0,9694	3,7402
ε=0.01	24,625	23	1,1873	2,0871	24	23	0,0349	0,3861
ε=0.025	12,25	11,625	1,603	0,8851	13	11	0,316	0,2322
ε=0.05	7	6,875	0,9797	0,3758	7	7	0,2441	0,0727
ε=0.1	4,75	4,625	0,5454	0,2745	5	5	0,1121	0,0517



ε	BOA*$_ε$ num$_{avg}$ sols	PPA* num$_{avg}$ sols	BOA*$_ε$ time$_{avg}$	PPA* time$_{avg}$	BOA*$_ε$ num$_{med}$ sols	PPA* num$_{med}$ sols	BOA*$_ε$ time$_{med}$	PPA* time$_{med}$
ε=0	108,562	108,5625	0,498	1,4158	92	92	0,1576	0,5824
ε=0.01	12	11,3125	0,259	0,1376	12	11	0,0856	0,0521
ε=0.025	6,1875	5,875	0,177	0,0681	6	6	0,0593	0,0249
ε=0.05	4,0625	3,6875	0,1063	0,0381	4	4	0,0296	0,0174
ε=0.1	2,25	2,25	0,035	0,0179	2	2	0,0232	0,0073



*BOA*ε vs PPA**

PPA*

- Λιγότερες επαναλήψεις
- Ακριβότερες λειτουργίες

- *ε>0*
- *Μεγάλα (υπο)γραφήματα*
- *Πολλές λύσεις*

BOA*

- Περισσότερες επαναλήψεις
- Φθηνές λειτουργίες

- *ε=0*
- *Μικρά (υπο)γραφήματα*
- *Λίγες λύσεις*

Ευχαριστώ για τον χρόνο και την προσοχή σας!

Νταλαγιώργος Αχιλλέας