A Distributed Augmenting Path Approach for the Bottleneck Assignment Problem

Transactions on Automatic Control

Additional Material

[*]Mitchell Khoo, [†]Tony A. Wood, [‡]Chris Manzie, [§]Iman Shames

May 31, 2022

[*] Department of Electrical and Electronic Engineering, The University of Melbourne, Australia; Email: khoom1@student.unimelb.edu.au

[†] Sycamore, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland; Email: tony.wood@epfl.ch

[‡] Department of Electrical and Electronic Engineering, The University of Melbourne, Australia; Email: manziec@unimelb.edu.au

[§] CIICADA Lab, School of Engineering, Australia National University, Australia; Email: iman.shames@anu.edu.au

Function PruneBAP with AugDFS, from agent's perspective

Input: Agent i, initial matched task m_i, agent's edge set $\mathcal{E}_i$, weight of edges $\mathcal{W}_i$.
Output: Final matched task m_i.

```
matching_exists ← True
$\bar{\mathcal{E}}_i \leftarrow \mathcal{E}_i$
while matching_exists do
    Find edge with largest weight in $\bar{\mathcal{E}}_i$
    Let the edge and the weight be a tuple $(e_i, w_i)$
    for $d \in D$ do
        for $k \in N(\mathcal{G}_C, i)$ do                              ▷ Neighbours of $i$
            Collect $(e_k, w_k)$
        end for
        Set the new $(e_i, w_i)$ to be the one containing the largest weight from the collect tuples
    end for
    $f_i$ = False                                                       ▷ $i$ is unexplored
    $\nu_i \leftarrow m_i$
    search_complete ← False
    Let $(\bar{i}, \bar{j}) = e_i$
    $t \leftarrow \bar{j}$
    $FILO \leftarrow t$
    while ¬search_complete do
        Check existence of edges $(i, t) \in \bar{\mathcal{E}}_i$
        If edge exists, set tuple $out_i = (m_i, w(i, m_i))$, or $out_i = (\hat{b}$,tiebreaking identifier) if $i$ is free, or else set it to $out_i = \emptyset$
        for $d \in D$ do
            for $k \in N(\mathcal{G}_C, i)$ do
                Collect $out_k$
            end for
            Set the new $out_i$ to be the one with smallest weight from the collect tuples, or $out_i = (\hat{b}$,tiebreaking identifier) if one of the collected tuples shows a free agent exists.
        end for
        if $out_i = \emptyset$ and $t = \bar{j}$ then                   ▷ No remaining agents
            search_complete ← True
            matching_exists ← False
        else if $out_i = \emptyset$ and $t \neq \bar{j}$ then           ▷ $t$ has no children
            Check and remove last element in $FILO$, $t^*$
            $t \leftarrow t^{**}$, where $t^{**}$ is the new last element in $FILO$ after removal of $t^*$
            if $m_i == t^*$ then
                $\nu_i \leftarrow t^*$
            end if
        else if $out_i = (\hat{b}$,tiebreaking identifier) then         ▷ Free agent found
            if $i$ matches the tiebreaking identifier then
                $\nu_i \leftarrow t$
            end if
            search_complete ← True
        else                                                            ▷ Explore next agent
            if $i$ matches the tiebreaking identifier then
                $\nu_i \leftarrow t$
                $f_i \leftarrow$ True                                   ▷ Mark $i$ as explored
            end if
            $t \leftarrow m_k$ from the current saved $out_i = (m_k, w(k, m_k))$
            Append $t$ to $FILO$
        end if
    end while
    $m_i = \nu_i$
end while
return $m_i$
```