Partial Cooperative Multi-Agent Periodic Indivisible Resource Allocation

Yuval Gabai Schlosberg and Roie Zivan

Ben-Gurion University of the Negev

AAMAS2021 ONLINE, MAY 3–7

1. Periodic Indivisible Resource Allocation (PIRA)

- A set of n Agents $A = \{A_1, A_2, \dots A_n\}$.
- A sent of m indivisible resources $R = \{r_1, r_2, \dots r_m\}$.
- An ordered set of atomic time units $\{t_1, t_2, \dots t_H\}$.
- H – the horizon (largest time unit considered).
- $<A_i, r_j, t_k>$ - *The assignment of* r_j *to* A_i *at time* t_k.
- *A complete assignment is a set of mH such triplets in which each combination of* r_j, *and* t_k *appears exactly once.*

2. PIRA as an ADCOP

- Variables- represent room allocations.
- The number of variables an agent holds equals the maximal number of rooms that can be allocated to the agent in the time interval.
- The utility derived by an agent from an allocation:

$$c = [\langle A_{i_1}, r_{j_1}, t_{k_1}\rangle, \dots, \langle A_{i_q}, r_{j_q}, t_{k_q}\rangle, u_{ic}]$$

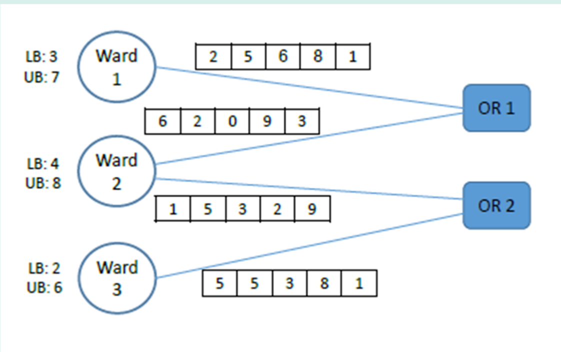
- The utility that A_i *derives from an allocation is the sum of utilities she derives from all constraints she is involved in.*
- *The utility of an allocation (a solution) is the sum of personal utilities of all agents.*

3. Hospital Operating Room Scheduling

Represented as a PIRA

- Agents represent wards.
- Variables represent room allocations.
- Values represent the room allocated to the ward at a specific time.
- Unary constraints represent the benefit for the ward.
- Binary hard constraints prevent conflicts such as assigning the same room to two departments on the same day.
- Cardinal constraints represent whether the amount of rooms allocated to a ward, addresses its needs.

4. Example



An example including:

- Three wards.
- Two Operating Rooms.
- For each ward the vectors specify the preferences over the allocations of the rooms to it in every day of the week.
- The example also includes an upper and lower bound for each ward representing the minimal number of allocations the ward can accept and the maximal number of rooms it can use in the time interval (e.g., a five working days week).

5. Partially Cooperative PIRA Algorithm

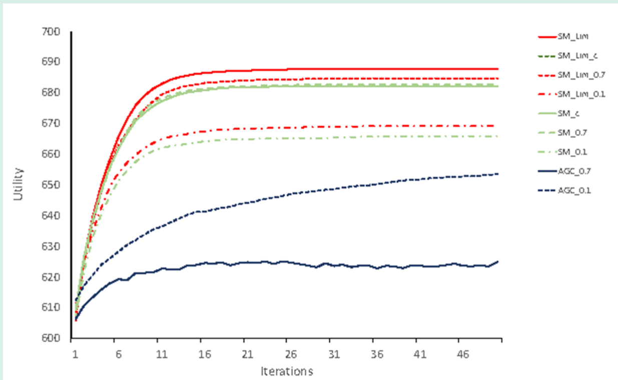
Algorithm 4 SM_AGC_PIRA

input: $baseLineAlloc_i$, $baseLineCost_i$, λ_i and Ω_i

$alloc \leftarrow baseLineAlloc_i$;
$\mu_i \leftarrow baseLineCost_i$;
$localView \leftarrow null$;
send($alloc$) to $N(i)$;
while stop condition not met **do**
  PHASE 1:
    Collect all *alloc* messages and update *localView*
    for each $A_j \in N(i)$ **do**
      $\pi_{i,j} \leftarrow preferences(A_j)$;
      send($\pi_{i,j}$) to A_j;
  PHASE 2:
    Collect all π messages;
    $\Pi_i \leftarrow \pi_{j \in N(i)} \cup preferences(A_i)$;
    $r_{iq} \leftarrow socialImprovingRequest(\Pi_i, \Omega_i)$;
    send($req_{iq}, socialGain_i$) to A_q;
  PHASE 3:
    Collect all $\langle r_{ji}, socialGain_j\rangle$ messages;
    $a_j \leftarrow$ agent in $N(i) \cup A_i$ with maximal socialGain s.t.
      c_i($localView$ after performing r_{ji}) $\leq \mu_i \cdot (1 + \lambda_i)$;
    send($Neg!$) to $N(i) \setminus a_j$;
  PHASE 4:
    Collect $Neg!$ messages;
    if did not receive $Neg!$ from A_q and from A_i & can improve
      then perform r_{iq};
    else if did not receive $Neg!$ from A_j
      then perform r_{iq};
    send($alloc$) to $N(i)$;

- Includes 4 synchronous phases.
- Agents propose trades and request resources.
- Agents accept trades and/or requests, if the result does not cause an unacceptable reduction of their utility or a breach of their bounds.

6. Experimental Evaluation



Algorithm Versions:

- AGC – no preference sharing among agents.
- SM – Socially Motivated (includes preference sharing).
- Lim – Using bounds.
- c – Complete cooperation.
- 0.1 / 0.7 – values of partial cooperation parameter λ.

The results indicate that:

- Socially motivated algorithms significantly out perform non preference sharing versions.
- The versions that use bounds are the most successful.

10. Conclusions

- Realistic applications often require model and algorithm adjustments.
- PIRA and the corresponding algorithms, can be used to represent and solve many realistic multi-agent optimization scenarios.
- There is a need to allow agents to balance between their own good and the organization's objectives.
- Socially Motivated Partial Cooperation successfully does so.