Coalition Formation for Task Allocation Using Multiple Distance Metrics (Student Abstract)

Tuhin Kumar Biswas[1], **Avisek Gupta**[2], **Narayan Changder**[2], **Redha Taguelmimt**[3], **Samir Aknine**[3], **Samiran Chattopadhyay**[4], **Animesh Dutta**[1]

[1]National Institute of Technology Durgapur, West Bengal, India.
[2] TCG CREST, Kolkata, West Bengal, India.
[3] LIRIS, Lyon 1 University, Lyon, France.
[4] Techno India University, West Bengal, India.
tuhin11biswas@gmail.com, avisek003@gmail.com, narayan.changder@tcgcrest.org, redha.taguelmimt@gmail.com, samir.aknine@univ-lyon1.fr, samirancju@gmail.com, animesh@cse.nitdgp.ac.in

Abstract

Simultaneous Coalition Structure Generation and Assignment (SCSGA) is an important research problem in multi-agent systems. Given n agents and m tasks, the aim of SCSGA is to form m disjoint coalitions of n agents such that between the coalitions and tasks there is a one-to-one mapping, which ensures each coalition is capable of accomplishing the assigned task. *SCSGA with Multi-dimensional Features* (SCSGA-MF) extends the problem by introducing a d-dimensional vector for each agent and task. We propose a heuristic algorithm called *Multiple Distance Metric* (MDM) approach to solve SCSGA-MF. Experimental results confirm that MDM produces near optimal solutions, while being feasible for large-scale inputs within a reasonable time frame.

Problem Formulation

Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a set of n heterogeneous agents and $\mathcal{T} = \{t_1, \ldots, t_m\}$ be a set of m independent tasks. The agents and tasks are each represented by d features representing problem attributes such as skills or resources of the agents, and likewise the skills or resources required to perform the tasks. Formally, the feature vector associated with agent $a_i \in \mathcal{A}$ is $S(a_i) = (s_1(a_i), \ldots, s_d(a_i))$ and with task $t_j \in \mathcal{T}$ is $S(t_j) = (s_1(t_j), \ldots, s_d(t_j))$. Here $s_i(.) \in \mathbb{R}$ is the feature value for the i-th skill or resource. Any coalition of agents $\mathcal{C}_j = \{a_{i_1}, \ldots, a_{i_{|\mathcal{C}_j|}}\} \subseteq \mathcal{A}$ is assigned a d-dimensional value $v(\mathcal{C}_j) = (\frac{1}{|\mathcal{C}_j|} \sum_{l=1}^{|\mathcal{C}_j|} S(a_{i_l}))$. Let $d_{\mathcal{M}}(x, y)$ be a measure of distance between two vectors x and y. In this paper we address the following problem of SCSGA with Multi-dimensional Features (SCSGA-MF):
Input: A tuple $\{\mathcal{A}, \mathcal{T}, \mathcal{S}(\mathcal{A}), \mathcal{S}(\mathcal{T}), v\}$; where $\mathcal{S}(\mathcal{A}) = \{S(a_1), \ldots, S(a_n)\}$, $\mathcal{S}(\mathcal{T}) = \{S(t_1), \ldots, S(t_m)\}$ and v provides the values for all $v(\mathcal{C}_j)$.
Output: A $\mathcal{CS} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ over $\mathcal{A}$ that minimizes $\sum_{j=1}^{m} d_{\mathcal{M}}(v(\mathcal{C}_j), S(t_j))$ to prioritize the assignment of agents that are closest to each task.
The following constraints must be satisfied by any two coalitions $\mathcal{C}_j, \mathcal{C}_k, \in \mathcal{CS} : \mathcal{C}_j, \mathcal{C}_k \neq \emptyset, \mathcal{C}_j \cap \mathcal{C}_k = \phi, \bigcup_{j=1}^{m} \mathcal{C}_j = \mathcal{A}$.

Proposed Algorithm

For SCSGA-MF, we design a heuristic to iteratively assign agents to tasks in two phases. In the first phase, based on an optimization criteria, agents are assigned until the requirements of all tasks are fulfilled. In the second phase, any remaining free agents are assigned to tasks in a balanced manner to minimize a second optimization criteria. Our heuristic operates by maintaining two lists: one for unfulfilled tasks (initially containing all tasks, denoted as $L(\mathcal{T}) = \{t_1, \ldots, t_m\}$, and the other for free agents (initially containing all agents, denoted as $L(\mathcal{A}) = \{a_1, ..., a_n\}$). The heuristic also precomputes a distance matrix, $D^c \in \mathbb{R}^{n \times m}$, where $D^c(i, j)$ provides a distance measure between vectors $S(a_i)$ and $S(t_j)$. The choice of a specific distance metric for D^c can influence the heuristic's performance by constraining it to certain agent and task distributions. To enhance the versatility of our Multiple Distance Metric (MDM) heuristic across a broader range of agent and task distributions (as proposed by (Zhang et al. 2022)), we compute and combine a set of $n_{\mathcal{M}}$ distance metrics $\{d_{\mathcal{M}}^{(1)}, ..., d_{\mathcal{M}}^{(n_{\mathcal{M}})}\}$ to form D^c. Thus, the combined distance matrix is computed as $D^c(i, j) = \sum_{k=1}^{n_{\mathcal{M}}} d_{\mathcal{M}}^{(k)}(S(a_i), S(t_j))$.

Given this initial setup, in the first phase, MDM iteratively assigns a single unique free agent to each unfulfilled task. These iterations continue until all tasks have been fulfilled. Let $U \in \{0, 1\}^{n \times m}$ be an assignment matrix, which allows each agent a_i to be assigned to exactly one task t_j by the imposed constraint $\forall i, \sum_{j=1}^{m} U(i, j) = 1$. In each iteration, every unfulfilled task $t_j \in L(\mathcal{T})$ is assigned a single unique free agent $a_i \in L(\mathcal{A})$, so as to optimize the following problem of the Partial Sum of the Distances of the Agent-Task Pairs (PSDATP): $\min \sum_{j=1}^{|L(\mathcal{T})|} \sum_{i=1}^{|L(\mathcal{A})|} U(i, j) D^c(i, j)$, subject to $\forall i, \sum_{j=1}^{m} U(i, j) = 1$. This problem is solved optimally by the Hungarian Algorithm (HA). If $|L(\mathcal{T})| \leq |L(\mathcal{A})|$, the HA identifies $|L(\mathcal{T})|$ agent-task pairs that minimize PSDATP, thus assigning a single unique free agent to each task. However, if in an iteration $|L(\mathcal{T})| > |L(\mathcal{A})|$, which can happen when there are fewer free agents left in $L(\mathcal{A})$ than unfulfilled tasks, the HA identifies $|L(\mathcal{A})|$ optimal agent-task pairs that minimize PSDATP, thus assigning the remaining agents to a subset of the unfulfilled tasks. After each iteration, $L(\mathcal{A})$ is updated to remove agents assigned in that iteration, and $L(\mathcal{T})$ is updated by removing tasks that have had all their requirements fulfilled. The first phase of MDM terminates when either $L(\mathcal{T}) = \emptyset$ or $L(\mathcal{A}) = \emptyset$. Thus, three possible outcomes can occur at the end of the first phase: (1) If $L(\mathcal{T}) \neq \emptyset$ and $L(\mathcal{A}) = \emptyset$, the MDM heuristic terminates with no free agents but unfulfilled tasks remaining. (2) If $L(\mathcal{T}) = \emptyset$ and $L(\mathcal{A}) = \emptyset$, then MDM terminates with no free agents and all tasks fulfilled. (3) If $L(\mathcal{T}) = \emptyset$ and $L(\mathcal{A}) \neq \emptyset$, then all tasks are fulfilled, yet free agents remain. The second phase of MDM only occurs when the third outcome arises. During this phase, the remaining free agents are assigned to tasks in a balanced manner until no free agents are left. In each iteration, a single free agent is assigned to a unique task by optimizing a similar problem of Partial Sum of the Distances to Agents: $\min \sum_{j=1}^{m} \sum_{i=1}^{|L(\mathcal{A})|} U(i,j) D^c(i,j)$, subject to $\forall i, \sum_{j=1}^{m} U(i,j) = 1$. Here the HA is used to identify $\min\{|L(\mathcal{A})|, m\}$ optimal assignments in each iteration. At the end of the second phase MDM thus terminates with no free agents and all tasks being fulfilled.

Once MDM terminates, all agents assigned to a task are identified as belonging to a coalition C_j, whose value is the mean of the agent vectors $v(C_j) = \frac{1}{|C_j|} \sum_{a_i \in C_j} S(a_i)$. The coalition structure *solution value* as per the multiple distance metrics can thus be calculated as $\sum_{j=1}^{m} \sum_{k=1}^{n_\mathcal{M}} d_\mathcal{M}^{(k)}(v(C_j), S(t_j))$. We also devise an Exact Algorithm (EA) for SCSGA-MF. EA explores all possible m-sized coalition structures using a brute-force approach and selects the coalition structure that produces the minimum *solution value*. Notice that the total number of possible coalition structures over n agents is denoted as B_n, where $\alpha n^{\frac{n}{2}} \leq B_n \leq n^n$. Hence, EA incurs a high computational cost, expressed as $O(\left\{ {n \atop m} \right\}) \equiv O(m^n)$. In comparison, MDM is more efficient: (i) If $m \geq n$, then only one iteration of the HA is required, incurring a cost of $O(n^2 m)$. (ii) If $m < n$, then a total of $\lceil n/m \rceil$ iterations are needed, and each iteration runs the HA for $O(nm^2)$. Thus, MDM runs in $O(\min\{\lceil n/m \rceil nm^2, n^2 m\}) = O(n^2 m)$ time.

Data Set	Solution Value		Run Time	
	MDM	EA	MDM	EA
UPD	51.94	46.83	0.002	100.64
NPD	111.24	111.10	0.001	102.09
SUPD	4.94	4.54	0.001	95.07
SNPD	3.86	3.79	0.001	96.6
FD	104.19	99.81	0.002	95.7
β	47.06	46.65	0.001	101.73

Table 1: Given 13 agents and 3 tasks. The average solution values, run times (in seconds) are shown for MDM and EA.

Experiments & Discussions: To evaluate the effectiveness of the MDM heuristic compared to EA, we follow the standard benchmarks used in (Präntare, Appelgren, and Heintz 2021). Specifically we use Uniform Probability Distribution (UPD), Normal (NPD), Sparse Uniform (SUPD), Sparse Normal (SNPD), F (FD), and Beta (β) to generate the values of the d-dimensional vectors (we consider $d = 5$) for the agents and tasks. For the set of $n_\mathcal{M} := 7$ distance metrics $\{d_\mathcal{M}^{(1)}, ..., d_\mathcal{M}^{(7)}\}$, we consider metrics from two categories. The first category consists of Minkowski metrics $\ell_p(x,y) = (\sum_{i=1}^{d} |x_i - y_i|^p)^{1/p}$, where we explore four such metrics with p values of 0.5, 1, 2, ∞. The three other metrics belong to the second category, defined on vector dot products. The first is the cosine distance $d_c(x,y) = 1 - \frac{x^T y}{||x||_2 \, ||y||_2}$, and the other two are polynomial distances $d_{p(c,d)}(x,y) = \alpha - ((1/e)x^T y + r)^e$, where $\alpha = max_{x,y}\{((1/e)x^T y + r)^e\}$, with $\{r, e\}$ as $\{0, 2\}$ and $\{1, 4\}$ respectively. Thus the set of seven distance metrics $\{d_\mathcal{M}^{(1)}, ..., d_\mathcal{M}^{(7)}\}$ are respectively $\{l_{0.5}, l_1, l_2, l_\infty, d_c, d_{p(0,2)}, d_{p(1,4)}\}$. To provide a consistent input space for all metrics to operate on, all input values are min-max scaled to [0,1]. Furthermore, each distance matrix computed from these metrics is normalized to [0,1] using their scalar min-max values, to facilitate their fair contribution towards D^c. Given 13 agents and 3 tasks, Table 1 shows the average solution values and run times of MDM and EA. EA can efficiently solve problems with up to 13 agents and 3 tasks quickly, but for 14 agents, its run time grows significantly, taking 4261 seconds for UPD. Hence, in Table 1, we report the results for MDM and EA with 13 agents and 3 tasks. Table 1 reveals that MDM's solution values closely match EA's, with minimal deviations, particularly for FD and UPD. Additionally, MDM consistently outperforms EA in terms of speed, finishing all cases in just 1 millisecond. Table 2 illustrates MDM's ability to efficiently handle large-scale inputs, accommodating up to 60,000 agents and 18,000 tasks within a reasonable time frame.

n	1e+4	2e+4	3e+4	4e+4	5e+4	6e+4
UPD	7	26	70	118	172	253
NPD	10	45	149	391	549	675
SUPD	8	28	67	114	169	253
SNPD	9	48	134	261	397	651
FD	9	59	128	277	481	785
β	8	30	68	114	172	247

Table 2: Average run times (in seconds) for increasing number of agents (n) and tasks ($m = \lceil 0.3 \times n \rceil$)

Conclusion: The efficacy of the MDM heuristic indicates that machine learning techniques which learn best fit parameterized distance metrics (Zhang et al. 2022) may succeed on the large-scale SCSGA-MF problem.

References

Präntare, F.; Appelgren, H.; and Heintz, F. 2021. Anytime Heuristic and Monte Carlo Methods for Large-Scale Simultaneous Coalition Structure Generation and Assignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11317–11324.

Zhang, Y.; Liu, X.; Liu, J.; Dai, S.; Zhang, C.; Xu, K.; and Zhu, E. 2022. Fusion Multiple Kernel K-means. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9109–9117.