DisLRP$_\alpha$: $\alpha$-approximation in Generalized Mutual Assignment

Katsutoshi Hirayama  
Kobe University  
5-1-1 Fukaeminami-machi, Higashinada-ku  
Kobe 658-0022, Japan  
hirayama@maritime.kobe-u.ac.jp

ABSTRACT

This paper presents a new distributed solution protocol, called DisLRP$_\alpha$, for the Generalized Mutual Assignment Problem (GMAP). The GMAP is a typical distributed combinatorial optimization problem whose goal is to maximize social welfare of the agents. Unlike the previous protocol for the GMAP, DisLRP$_\alpha$ can provide a theoretical guarantee on global solution quality. In DisLRP$_\alpha$, as with in the previous protocol, the agents repeatedly solve their local problems while coordinating their local solutions using a distributed constraint satisfaction technique. The key difference is that, in DisLRP$_\alpha$, each agent is required to produce a feasible solution whose local objective value is not lower than $\alpha$ ($0 < \alpha \leq 1$) times the local optimal value. Our experimental results on benchmark problem instances show that DisLRP$_\alpha$ can certainly find a solution whose global objective value is higher than that theoretically guaranteed. Furthermore, they also show that, while spending extra communication and computation costs, DisLRP$_\alpha$ can produce a significantly better solution than the previous protocol if we set $\alpha$ appropriately.

Categories and Subject Descriptors

1.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Coherence and coordination

General Terms

Algorithms, Performance

1. INTRODUCTION

Recently, we have proposed the Generalized Mutual Assignment Problem (GMAP) as a distributed formulation of the Generalized Assignment Problem (GAP) [4]. The GAP is a centralized NP-hard problem whose goal is to find the most profitable assignment of jobs to agents such that every job is assigned to exactly one agent and the assignment satisfies all of the resource constraints imposed on individual agents. The GMAP, on the other hand, can be considered a distributed problem, where the agents themselves, each of which has a set of jobs, try to solve the entire GAP while communicating with each other.

We have also presented DisLRP$_\alpha$ as a protocol for the GMAP that yields a solution providing a lower bound of the global optimal value [4]. In this protocol, the agents repeatedly solve their local problems while coordinating their local solutions using a distributed constraint satisfaction technique. Note that, unlike other protocols for similar problems [1, 3, 6], it is a pure distributed solution protocol, which can work without using any global entity (such as server or coordinator).

The drawback of DisLRP$_\alpha$ is solution quality. It exploits randomness in solving local problems in order to produce quick agreement between the agents on a global solution. By controlling the degree of randomness one can control the quality of the solution to some extent, but no theoretical guarantee is provided on that quality. This paper presents a new distributed solution protocol, called DisLRP$_\alpha$ for the GMAP that can provide a theoretical guarantee on global solution quality. DisLRP$_\alpha$ accepts a certain value of $\alpha$ ($0 < \alpha \leq 1$) as an input and can produce a global solution whose objective value is not lower than $\alpha$ times the global optimal value.

The remainder of this paper provides a formal definition of the GMAP followed by a new property, an outline of the new protocol, and a part of our experimental results.

2. GENERALIZED MUTUAL ASSIGNMENT PROBLEM

The goal of the GAP is to find the most profitable assignment of $n$ jobs to $m$ agents such that every job is assigned to exactly one agent and the assignment satisfies all of the resource constraints imposed on individual agents. It can be formulated as the following integer programming problem:

$$
\text{GAP} \quad \text{maximize} \quad \sum_{k \in A} \sum_{j \in J} p_{kj} x_{kj} \\
\text{subject to} \quad \sum_{k \in A} x_{kj} = 1, \quad \forall j \in J, \\
\sum_{j \in J} w_{kj} x_{kj} \leq c_k, \quad \forall k \in A, \\
x_{kj} \in \{0, 1\}, \quad \forall k \in A, \forall j \in J,
$$

where $A = \{1, \ldots, m\}$ is a set of agents; $J = \{1, \ldots, n\}$ is a set of jobs.
of jobs; \( p_{kj} \) and \( w_{kj} \) are the profit and amount of resource required, respectively, when agent \( k \) selects job \( j \); \( c_k \) is the capacity (amount of available resource) of agent \( k \). \( x_{kj} \) is a decision variable whose value is set to 1 when agent \( k \) selects job \( j \) and 0 otherwise.

The Lagrangian relaxation problem for the GAP, denoted as \( \mathcal{LGAP}(\mu) \), is obtained by dualizing the assignment constraints (1) of \( \mathcal{GAP} \):

\[
\mathcal{LGAP}(\mu) \quad (\text{decide } x_{kj}, \forall k \in A, \forall j \in J):
\]

\[
\begin{align*}
\text{max.} & \quad \sum_{k \in A} \sum_{j \in J} p_{kj} x_{kj} + \sum_{j \in J} \mu_j \left( 1 - \sum_{k \in A} x_{kj} \right) \\
\text{s. t.} & \quad \sum_{j \in J} w_{kj} x_{kj} \leq c_k, \forall k \in A, \tag{4} \\
& \quad x_{kj} \in \{0, 1\}, \forall k \in A, \forall j \in J, \tag{5}
\end{align*}
\]

where \( \mu_j \) is a real-valued parameter called a Lagrange multiplier and the vector \( \mu = (\mu_1, \mu_2, \ldots, \mu_n) \) is called a Lagrange multiplier vector. It is known that for any value of \( \mu \) the optimal value of \( \mathcal{LGAP}(\mu) \) provides an upper bound of the optimal value of \( \mathcal{GAP} \).

Since, in \( \mathcal{LGAP}(\mu) \), the objective function is additive over the agents and the constraints (4) are separable over the agents, this maximization can be achieved by solving the following subproblems, each belongs to agent \( k \):

\[
\mathcal{LGMP}_k(\pi_k(\mu)) \quad (\text{decide } x_{kj}, \forall j \in R_k):
\]

\[
\begin{align*}
\text{max.} & \quad \sum_{j \in R_k} p_{kj} x_{kj} + \sum_{j \in R_k} \mu_j \left( 1 - \sum_{j \in R_k} x_{kj} \right) \\
\text{s. t.} & \quad \sum_{j \in R_k} w_{kj} x_{kj} \leq c_k, \\
& \quad x_{kj} \in \{0, 1\}, \forall j \in R_k,
\end{align*}
\]

where \( R_k \) is a set of jobs that may be assigned to agent \( k \) and \( S_j \) is a set of agents to whom job \( j \) may be assigned. We can assume that \( S_j \neq \emptyset \) because a job with an empty \( S_j \) does not have to be considered. \( \pi_k(\mu) \) indicates a projection of \( \mu \) over the jobs in \( R_k \).

In the GMAP, the agents themselves try to generate an optimal solution of the GAP from a non-optimal (and sometimes infeasible) solution of it. We have already shown that distributed solution is possible by exploiting the following properties on the relation between the decomposed subproblems and the global problem [4].

**Property 1.** For any value of \( \mu \), the total sum of the optimal values of \( \{ \mathcal{LGMP}_k(\pi_k(\mu)) | k \in A \} \) provides an upper bound of the optimal value of \( \mathcal{GAP} \).

**Property 2.** For some value of \( \mu \), if all of the optimal solutions to \( \{ \mathcal{LGMP}_k(\pi_k(\mu)) | k \in A \} \) satisfy the assignment constraints (1) of \( \mathcal{GAP} \), then these optimal solutions constitute an optimal solution to \( \mathcal{GAP} \).

This paper presents the following new property resulting the new protocol. It must be noted that this is a generalization of Property 2.

**Property 3.** For some value of \( \mu \), if all of the feasible solutions to \( \{ \mathcal{LGMP}_k(\pi_k(\mu)) | k \in A \} \) satisfy the assignment constraints (1) of \( \mathcal{GAP} \), and their objective values are not lower than \( \alpha (0 < \alpha \leq 1) \) times the respective optimal values, then these feasible solutions constitute a feasible solution to \( \mathcal{GAP} \) whose objective value is not lower than \( \alpha \) times the optimal value.

**Proof.** Let an objective value of a feasible solution to \( \mathcal{LGMP}_k(\pi_k(\mu)) \) be \( v_k(\mu) \), the optimal value of \( \mathcal{LGMP}_k(\pi_k(\mu)) \) be \( \text{opt}_k(\mu) \), and the optimal value of \( \mathcal{GAP} \) be \( \text{OPT} \). Obviously, those feasible solutions constitute a feasible solution to \( \mathcal{GAP} \) since they satisfy all constraints of \( \mathcal{GAP} \). Furthermore, the objective value of such feasible solution to \( \mathcal{GAP} \) is equal to \( \sum_{k \in A} v_k(\mu) \). Since \( v_k(\mu) \geq \alpha \times \text{opt}_k(\mu) \) for any agent \( k \), we obtain

\[
\sum_{k \in A} v_k(\mu) \geq \sum_{k \in A} \alpha \times \text{opt}_k(\mu) = \alpha \sum_{k \in A} \text{opt}_k(\mu) \geq \alpha \times \text{OPT},
\]

where we use Property 1 for the second inequality.

3. PROTOCOL

### 3.1 Basic Methods

An overall behavior of the basic protocol is that the agents start with \( t = 0 \) and \( \mu^{(0)} = (0, \ldots, 0) \) and alternate solving primal and dual problems while performing local communication with their respective neighbors. More specifically, each agent \( k \) behaves as follows:

Step 1: sets \( t = 0 \) and \( \mu^{(0)} = (0, \ldots, 0) \).

Step 2: solves \( \mathcal{LGMP}_k(\pi_k(\mu^{(t)})) \) /* primal problem */.

Step 3: exchanges solutions with her neighbors.

Step 4: updates the Lagrange multiplier vector from \( \mu^{(t)} \) to \( \mu^{(t+1)} \) /* dual problem */.

Step 5: increases \( t \) by one and goes to Step 2.

The loop between Steps 2 and 5 is called a round and stops when all of the assignment constraints are satisfied or a pre-specified upper bound of rounds is reached.

The primal problem, \( \mathcal{LGMP}_k(\pi_k(\mu^{(t)})) \) with a specific Lagrange multiplier vector in Step 2, is virtually the \( 0-1 \) knapsack problem, which is known to be an “easier hard” NP-hard problem [2]. Neighbors of agent \( k \), mentioned in Step 3, are informally stated as a set of agents with whom agent \( k \) shares assignment constraints. They can be formally described as \( \bigcup_{j \in R_k} S_j \setminus \{k\} \) [4].

To solve the dual problem an agent uses a subgradient optimization method. In this method, agent \( k \) first computes subgradient \( g_j \) for each job \( j \) in \( R_k \) by

\[
g_j = 1 - \sum_{i \in S_j} x_{ij}.
\]

Intuitively, \( g_j \) means the gap between the number of agents required for job \( j \) (one in this case) and the number of agents that currently select this job. Then, in Step 4, agent \( k \) updates \( \mu_j \) for each job \( j \) in \( R_k \) by

\[
\mu_j^{(t+1)} - \mu_j^{(t)} = -\frac{f(t)}{|S_j|} g_j,
\]

where \( f(t) \) is a step length, a positive scalar parameter that may vary depending on discrete time \( t \), and \( |S_j| \) is the number of agents involved in job \( j \). According to this rule, \( \mu_j \)
gets higher when job \( j \) attracts too many agents and lower when it attracts too few agents. Namely, \( \mu_j \) can be viewed as the price of job \( j \).

By repeating Steps 2-5 we can expect that an upper bound of the global optimal value of GAP gradually decreases, and thus the agents eventually reach to a state that is close to a global optimal solution. In the meantime, if all of the assignment constraints are satisfied, the protocol can be terminated because this fact indicates the agents have reached to a global optimal solution. Note that each agent can detect the fact by using the termination detection technique of the distributed breakout algorithm [5].

### 3.2 New Protocol
DisLRP\(_a\) relies on Property 3, which says that a sufficient condition for a global solution to have at least the quality of \( \alpha \) is: 1) all of the feasible solutions to \( \{LGMP_k(\pi_{(\mu^{(t)})}) | k \in A \} \) satisfy the assignment constraints, and 2) the objective values of their feasible solutions are not lower than \( \alpha \) times the optimal value, \( \pi^{(t)} \). This condition can be achieved by the new protocol, where each agent \( k \) repeats the following steps until all of the assignment constraints are satisfied:

- **Step 1:** sets \( t = 0 \) and \( \mu^{(0)} = (0, \ldots, 0) \),
- **Step 2:** finds a feasible solution to \( LGMP_k(\pi_{(\mu^{(t)})}) \) whose objective value is not lower than \( \alpha \) times the optimal value,
- **Step 3:** exchanges solutions with her neighbors,
- **Step 4:** updates the Lagrange multiplier vector from \( \mu^{(t)} \) to \( \mu^{(t+1)} \),
- **Step 5:** increases \( t \) by one and goes to Step 2.

Note that this is different from the basic protocol only in Step 2. While the basic protocol requires an optimal solution in Step 2, the new protocol can relax this requirement.

In Step 2 of the new protocol, we use a simple method for finding such a feasible solution. In this method each agent generates a “skewed” subproblem by randomly disturbing current objective coefficients. An agent exploits an optimal solution to this “skewed” subproblem as a feasible solution to a “true” subproblem. More specifically, if an optimal solution to a skewed subproblem has a “true” objective value that is not lower than \( \alpha \) times the “true” optimal value, an agent uses this “skewed” optimal solution; otherwise, uses the “true” optimal solution.

### 4. EXPERIMENTS
We made experiments to compare DisLRP\(_a\) and the family of DisLRP\(_a\) (DisLRP\(_{\alpha,90}\), DisLRP\(_{\alpha,95}\), and DisLRP\(_{\alpha,99}\)) on some GMAP instances, which were derived from GAP benchmark instances (gap11, gap12, gapa, and gapb from the OR-Library). These experiments were conducted on a discrete event simulator that simulates the behavior of the protocols. In the simulator we used ILOG CPLEX ver 8.1 for each agent to solve a local 0-1 knapsack instance. We made 10 runs for each instance and a run was cut off at 5000 rounds. Figure 1 shows the quality of global solutions found by each protocol for the gap11 instances. In this figure, each vertical line represents a range of quality over 10 runs and a symbol on the line represents their mean.

![Figure 1: Range of solution quality for gap11](image)

For every instance we can see a clear trend that DisLRP\(_a\) can find a global solution with higher quality than DisLRP\(_L\). We should point out that the quality actually achieved by DisLRP\(_a\) is higher than that theoretically guaranteed. Furthermore, we can observe that increasing \( \alpha \) actually leads to finding global solutions with higher quality and as a result the variance of solution quality can be naturally reduced.

On the other hand, we observed that DisLRP\(_a\) obviously spent much more rounds to find a global solution. This indicates that total communication and computation costs for DisLRP\(_a\) are much higher than those for DisLRP\(_L\). However, we consider that these are inevitable costs for finding a global solution with higher quality. One should select a protocol that provides the best balance between solution quality and solution costs depending on one’s requirement.

### 5. CONCLUSIONS
This paper presented DisLRP\(_a\) for the GMAP, which can provide a theoretical guarantee on global solution quality. DisLRP\(_a\) can certainly find a solution whose global objective value is higher than that theoretically guaranteed and, while spending extra communication and computation costs, it can produce a significantly better solution than DisLRP\(_L\) if we set \( \alpha \) appropriately.

### 6. REFERENCES