Collaborative Job Processing on a Single Machine – A Multi-Agent Weighted Tardiness Problem

(Extended Abstract)

Fabian Lang
Institute of Computer Science
Helmut Schmidt University
Holstenhofweg 85
Hamburg, Germany
fabian.lang@hsu-hamburg.de

Andreas Fink
Institute of Computer Science
Helmut Schmidt University
Holstenhofweg 85
Hamburg, Germany
andreas.fink@hsu-hamburg.de

ABSTRACT

We present a multi-agent variant of the Single Machine Total Weighted Tardiness Problem with Sequence-Dependent Setup Times. Since, i.a., agents have an incentive to lie, central planning is not feasible and decentralized methods such as automated negotiations are needed. Hereto, we propose and evaluate an iterative quota-based negotiation protocol.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]:
Distributed Artificial Intelligence - Multiagent systems

General Terms

Algorithms, Economics, Experimentation

Keywords

Automated negotiation, Autonomous agents, Machine scheduling, Interorganizational system, Agent coordination

1. PROBLEM DESCRIPTION

In the following, we discuss the problem of single machine scheduling with several self-motivated, non-cooperative agents – with applications such as allocation of processing power, satellite data transmission , or terminal scheduling in harbors – and present a negotiation protocol constituting a coordination mechanism to find beneficial agreements. The *Multi-Agent Single Machine Total Weighted Tardiness Problem with Sequence-Dependent Setup Times* (MA-SMTWTP-SDST) is a scheduling problem where a set of jobs $\mathcal{J} = \{1, ..., j, ..., n\}$ has to be processed by a single machine, which can process only a single job at a time. Each job j is assigned to an agent i of the set $\mathcal{I} = \{1, ..., i, ..., m\}$ by an assignment variable a_j. The agents aim at minimizing their individual total weighted tardiness TWT_i (see (1b)). Each job comprises a processing time $p_j(> 0)$, a weight of relative importance $w_j(> 0)$, and a due date D_j. A setup time $s_{k,j}$ occurs between a job j and its preceding job k (see (3)) and c_j denotes the completion time of job j. If a job is not finished before its due date, a tardiness T_j arises (see (2)). The objective of the problem is to find a job sequence $\pi = \{\pi_1, \ldots, \pi_n\}$ (with π_j as processing position) minimizing the collective total weighted tardiness TWT (see (1a)). Lawler [4] shows that the centralized problem (without SDST) is already strongly $\mathcal{NP}$-hard. Since the agents have an incentive to lie (e.g., by declaring their jobs as more important than they really are), revealed information about due dates as well as job weights are worthless and not utilizable. Hence, centralized planning is not feasible here.

$$\min \quad \sum_{i=1}^{m} TWT_i \tag{1a}$$

$$\min \quad TWT_i = \sum_{j \in \mathcal{J} | a_j = i} w_j T_j \ , \ \forall i \in \mathcal{I} \tag{1b}$$

$$T_j = \max\{c_j - D_j; 0\} \tag{2}$$

$$c_j = \sum_{k \in \mathcal{J} | \pi_k \leq \pi_j} s_{\tilde{k}|(\pi_{\tilde{k}} = \pi_k - 1),k} + p_k \tag{3}$$

2. THE PROPOSED PROTOCOL

Here, we present our proposed negotiation protocol for multi-agent coordination (see algorithm 1). The basic idea is that agents overcome local optima by accepting deteriorating contract proposals [2][3]. The protocol is generic. At first, a mediator generates a random initial contract c_o^* that represent the active contract (=current draft). In every iteration t, an acceptance quota p_t for the set of proposals is determined, which declines over time (first round: p_o; last round: $p_{T-1} \stackrel{!}{=} \frac{1}{L}$). Subsequently, the mediator creates $L-1$ mutations $c_t^{'}$ of the active contract c_t^*. Those mutations and the active contract constitute the set of contract proposals $\mathcal{C}^{'}$. Afterwards, the agents decide whether to accept (=1) or reject (=0) the proposals, but have to accept at least $q_t(= p_t * L)$ contracts. We suppose that they accept the $q_t * L$ best contracts as well as possible improving contracts. The mediator selects one randomly from the overall accepted contracts $\mathcal{C}^c$; if there is none ($\mathcal{C}^c = \emptyset$), the active contract remains for the next iteration. Thereafter, the process starts over and new proposals are generated using the (new) active contract. Finally, after T iterations, the last active contract c_{T-1}^* becomes the final contract c^*.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

Algorithm 1 An Iterative, Quota-Based Protocol

$c_0^* \leftarrow GenerateInitialContract()$
for $t = 0, 1, \ldots, T-1$ **do**
  $p_t \leftarrow p_0 * \beta^t$; $q_t \leftarrow p_t * L$
  $\mathcal{C}^c \leftarrow \emptyset$; $\mathcal{C}' \leftarrow \{c_t^*\}$
  for $l = \{1, 2, \ldots, L-1\}$ **do**
    $c_t' \leftarrow Mutate(c_t^*)$
    $\mathcal{C}' \leftarrow \mathcal{C}' \bigcup \{c_t'\}$
  end for
  for all $i \in \mathcal{I}$ **do**
    $\overrightarrow{Z_i} \leftarrow AcceptOrReject(\mathcal{C}', q_t)$
  end for
  for all $c_t' \in \mathcal{C}'$ **do**
    if $\sum_{i \in \mathcal{I}} Z_i[c_t'] = m$ **then**
      $\mathcal{C}^c \leftarrow \mathcal{C}^c \bigcup \{c_t'\}$
    end if
  end for
  if $\mathcal{C}^c = \emptyset$ **then**
    $c_{t+1}^* \leftarrow c_t^*$
  else
    $c_{t+1}^* \leftarrow RandomlySelect(\mathcal{C}^c)$
  end if
end for
$c^* \leftarrow c_{T-1}^*$

3. COMPUTATIONAL RESULTS

For the evaluation, we have used the 120 problem instances of the SMTWTP-SDST benchmark library from [1] and assigned the jobs to multiple agents in sequence. Since there are loose and tight problem sets, we have normalized the results: 100% represents the best found value of a problem set in the respective simulation data set.

There are three adjustable parameters: the number of iterations T, the number of proposals $L-1$, and the initial acceptance ratio p_0. Another decision parameter is the mediator's way of generating proposals such as, firstly, shifting a single item to another position in the sequence and, secondly, swapping two items' positions. Table 1 shows the results of different parameterizations of p_0 and L assuming $m = 5$ agents and 100,000 iterations.

Table 1: Configuration of the Quotas

Mutation	*Shifting*			*Swapping*		
L	10	25	50	10	25	50
$p_0 = 10\%$	429%	259%	246%	288%	159%	157%
$p_0 = 33\%$	156%	151%	148%	126%	116%	111%
$p_0 = 50\%$	138%	125%	134%	121%	112%	112%
$p_0 = 67\%$	129%	127%	127%	127%	112%	112%
$p_0 = 90\%$	134%	123%	127%	127%	113%	116%

As shown, the quota protocol needs a sufficiently high absolute value of accepted contracts to succeed. The results tend to be better if there are more proposals as well as higher demanded quotas so we have chosen $\{L = 25; p_0 = 0.67\}$ for the remainder of the paper. Concerning the mutation method, swapping appears to perform better.

Regarding the number of agents, table 2 shows that the TWT increases with more agents, although the jobs are the very same. We trace this finding to a more difficult coordination process between the agents.

Table 3 shows a comparison between the quota-rule and free decision making for different negotiation lengths. The performance is slightly increasing with a rising number of iterations (but converging) and the quota-rule outperforms free decision making by far.

Table 2: Number of Agents

Agents	2	3	4	5	10	15	20
Tardiness (%)	102	114	119	126	135	141	150

Table 3: Iterations and Comparison

Iterations	10K	50K	100K	500K	1,000K
Quota	122%	110%	105%	108%	106%
No Quota	252%	244%	237%	247%	241%

Finally, we have approximated the Pareto frontier using a multi-objective simulated annealing procedure (MOSA, see [5]) with 5,000 runs. Figure 1 depicts the history of an exemplary negotiation between two agents over one million negotiation rounds as well as the Pareto frontier. The negotiation moves intensively through the contract space. At the end, the negotiation has converged and is moving up and down in the neighborhood of the Pareto frontier. The protocol finds even better results than the centralized MOSA.

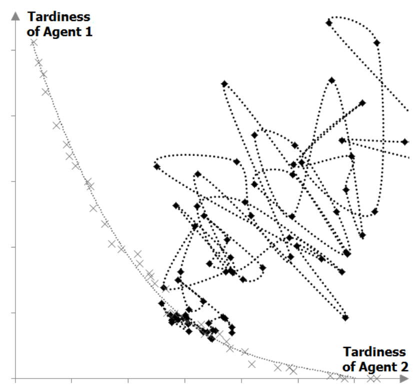

Figure 1: Neg. History & Approx. Pareto Frontier

4. REFERENCES

[1] V. Cicirello. Weighted tardiness scheduling with sequence-dependent setups: a benchmark library. Technical Report, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2003. *http://loki.stockton.edu/∼cicirelv/benchmarks.html*

[2] A. Fink. Supply chain coordination by means of automated negotiations between autonomous agents. In B. Chaib-draa and J. Müller, editors, *Multiagent based Supply Chain Management*, pages 351–372. Springer, Berlin / Heidelberg, Germany, 2006.

[3] M. Klein, P. Faratin, H. Sayama, and Y. Bar-Yam. Negotiating complex contracts. *Group Decision and Negotiation*, 12(2):111–125, 2003.

[4] E. Lawler. A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1:331–342, 1977.

[5] E. L. Ulungu and J. Teghem. Multi-objective combinatorial optimization problems: a survey. *Journal of Multi-Criteria Decision Analysis*, 3(2):83–104, 1994.