

# マルチエージェントカレンダー情報共有を用いた 論文投稿支援システム

磯村 厚誌 大園 忠親 新谷 虎松

名古屋工業大学 知能情報システム学科

e-mail: {isomura,ozono,tora}@ics.nitech.ac.jp

## 1 はじめに

個人のスケジュールを管理するためのツールの一つとして、カレンダーソフトが挙げられる。近年、一般に普及しているカレンダーソフトの中にも、ネットワークを介した情報共有が可能なものがあり、グループ単位でのスケジュール管理も可能になった。しかし、それらは単にコミュニケーションを支援するだけのものが多く、複雑な作業にかかる負担はいまだ大きなものである。

本稿では、カレンダーの新規イベントを計画する際に、既存のイベントの再スケジュールリングを視野に入れた日程スケジュールリングを提案する。より具体的な応用として、論文投稿の際の、研究内容に関するミーティングスケジュールの立案支援を行う。この機能を中心に、論文投稿を支援するマルチエージェントカレンダーシステムを実装した。我々の論文投稿支援システムでは、分散環境における日程スケジュールリングの問題を分散重み付き制約充足問題 (DVSCP) として定式化する。また、再スケジュールリング後の混乱を抑制するために、ユーザが意図的に追加する制約の他にシステムからも制約の追加を行う。

## 2 分散カレンダーシステム

### 2.1 システム概要

本システムは、図1に示すように、公開されたカレンダー情報の管理を行なう公開カレンダーサーバーと、各個人のカレンダーシステムで構成される。それぞれ、公開カレンダー管理エージェント(サーバーエージェント)と個人カレンダー管理エージェント(個人エージェント)によって操作される。個人エージェントはサーバーエージェントに対しカレンダー情報を公開し、サーバーエージェントは、公開された全てのカレンダー情報から各個人エージェントの要求に対応したカレンダーを編集、提供する。また、各個人エージェント間での通信も可能であり、複数の参加者を持つイベントについて非公開で計画を立てることが可能である。

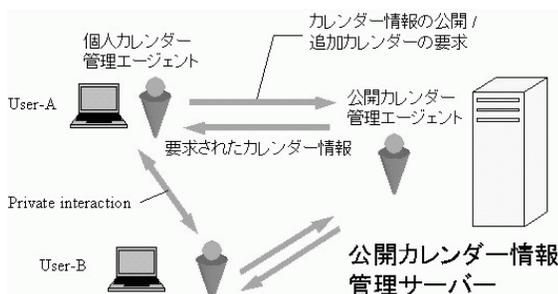


図 1: システム概要

Multi Agent Calendar System for Paper Submission Scheduling.

Atsushi Isomura, Tadachika Ozono, Toramatsu Shintani

Dept. of Intelligence and Computer Science, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555

### 2.2 カレンダーによる論文投稿支援

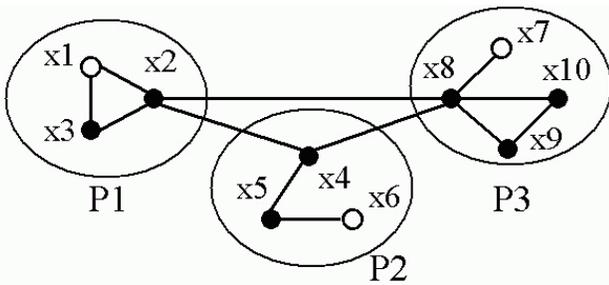
本研究における具体的な目的は、主に論文投稿前の、研究に関するミーティング日程スケジュールリングを行なうことである。また、締め切りや開催日などの日程に関する情報も、その情報に気づいたユーザがカレンダーを公開することで関係者全員に伝えられる。日程に関する情報については、ホームページなどの情報源が特定できる場合には、サーバーエージェントに情報を収集させることも可能である。ミーティング日程のスケジュールリングは、分散重み付き制約充足問題 (DVSCP) として考え、個人エージェント間での解決を試みる。

## 3 分散重み付き制約充足問題

制約が強過ぎて(過制約)解が存在しない場合に、問題を扱えるように CSP を拡張した枠組の1つが重み付き CSP である。重み付き CSP では、各制約に重みを与え、充足できない制約の重みの集合を最小にする解を探索する。この重み付き CSP をさらに分散環境へ拡張した枠組が、分散重み付き CSP [2] である。この CSP は、重み付き CSP が複数のエージェント上に分散された問題である。分散重み付き CSP は、 $DVP = (P, P_I)$  によって定義する。 $P = P_1, \dots, P_m$  は重み付き CSP の集合である。 $P$  の各重み付き CSP  $P_i = (X_i, D_i, C_i, S, \varphi_i)$  は、変数集合  $X_i$ 、変域集合  $D_i$ 、制約集合  $C_i$ 、評価構造  $S = (E, \otimes, \succ)$  および評価関数  $\varphi_i : C \rightarrow E$  を持つ。ここで、 $E$  は評価値の集合、 $\succ$  は  $E$  上で全順序である。 $\otimes$  は評価値を総計するオペレーションである。 $P_I = (X_I, D_I, C_I, S, \varphi_I)$  は複数のエージェントに関する重み付き CSP で、変数集合  $X_I (\bigcup_{i=1}^m X_i$  の部分集合)、変域集合  $D_I$ 、制約集合  $C_I$ 、評価構造  $S$ 、および評価関数  $\varphi_I$  を持つ。 $A$  は全変数への値割当てとする。分散重み付き CSP では、制約  $c$  に関する  $A$  の評価値を  $\varphi(A, c)$  とすると、全変数の値割当ての評価値は、 $\varphi(A) = \otimes_{c \in C} \varphi(A, c)$  と定義する。ここで、 $C = (\bigcup_{i=1}^m C_i) \cup C_I$  は全制約集合である。

## 4 分散カレンダーにおける ミーティングスケジュールリング

分散重み付き CSP では、エージェント間で制約を持つ変数以外の変数は各エージェント内で保持され、他のエージェントが正確に情報を得ることが出来ない。このため、個人のカレンダーを統合して利用する場合にプライベートな情報をより確実に保護することが出来る。このような背景から、参考文献 [1] では会議日程のスケジュールリング問題に重み付き CSP を適用している。この会議日程スケジュールリングでは、既に決定している日程の変更は好ましくないとし、既存の日程を単に制約として扱っている。しかし本稿では、個人のカレンダーが自動的に再スケジュールリング可能なイベントを含むものとする。例えば、本論文投稿支援システムにおいて、個人のカレンダーには実験計画や資料作成の予定も記述されるものとし、制約として、現在の時刻  $ct$  から3日以内のイベントは変更不可、等の条件を持ち、この場合は3日後の時刻を  $et$  とし、 $determined([ct, et])$



● : public変数 ○ : private変数

図 2: 分散重み付き CSP の例

と表す. このような, イベントの基本的な性質を示す制約はシステムから各イベントに一括して与えられ, ユーザからの制約は各々の状況に合わせて追加される.

## 5 スケジューリングのプロセス

スケジュールを構成する要素であるイベントは  $e_{ij}$  (エージェント  $i$  のイベント  $j$ ) と表す.  $e_{ij}$  は, 開始日時  $st$ , 終了日時  $et$  ( $st \in T, et \in T, st < et$ ), および内容情報  $N_t$  の 3 つの要素で構成される. ここで,  $T$  は日時を表す離散値集合である. 制約  $c_{ij}$  はシステムに対応するフォーマットで記述し, イベント  $e_{ij}$  と制約  $c_{ij}$  にはユーザによって 1 から 9 の重みが付けられる. 提案されるイベント, 及び再スケジュールリングによる変更が可能なイベントは, 日程の候補になる時間帯  $T_s = \{t_1, t_2, \dots, t_n\} (t_i = [x, y] (x, y \in T, x < y))$  と制約の重みの閾値  $T_r \in [1, 7]$ , 及びイベントに関する情報  $N_t$  で構成される.

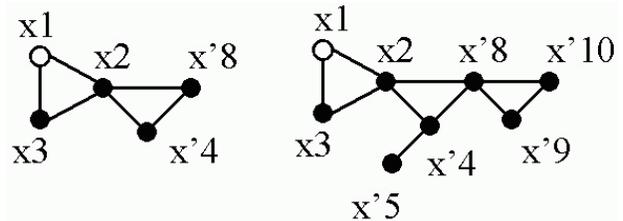
我々は, 日程スケジュールリング問題を分散重み付き CSP  $DP = (P, P_I)$  として定義した.  $P = P_1, \dots, P_m$  は各個人エージェントの重み付き CSP の集合である. エージェント  $i$  の重み付き CSP  $P_i = (X_i, D_i, C_i, S, \varphi_i)$  は, 各イベント  $e_{ij}$  の開始日時  $st$  と終了日時  $et$  を変数集合  $X_i$  とし, その領域集合  $D_i$  は日時集合  $T$ , 制約集合  $C_i$  はそれぞれのイベントについての制約の集合である. 評価構造  $S$  は,  $E = [0, 9], >=>, \otimes = +$  で, 評価関数  $\varphi$  はユーザによって与えられる.  $PI = (X_I, D_I, C_I, S, \varphi_I)$  はエージェント間の重み付き CSP である.  $X_I$  は複数エージェント間で共有される変数であり, 開始及び終了日時を表す. その領域集合  $D_I$  は日時集合  $T$  である.  $C_I$  はこれらのエージェント間に存在する制約の集合であり, 評価関数  $\varphi_I$  はスケジュールリングアルゴリズムの制約の閾値に基づく.

以上のように定式化した分散重み付き CSP の例を図 2 に示す.  $P_1, P_2, P_3$  はそれぞれ 1 つのエージェントが持つ重み付き CSP である. 図中の黒いノードはカレンダーにおいてパブリックな変数を表し, 白いノードがプライベートな変数を示す. そして各ノードを結ぶ辺がそれぞれの変数間の制約である. このとき, 複数のエージェントに渡る制約がエージェント間制約であり, その制約に関連する変数がエージェント間変数である. これらは必ず  $P_I$  にも含まれる.

我々は, DVCSP として定式化したミーティング日程のスケジュールリングに対し, greedy repair distributed method[2] を用いる. ただし, この手法が  $P_I$  を構成する際に,  $X_I$  としてエージェント間変数しか考慮しないのに対し, 我々は, システムのサーバーエージェントに公開されたイベントに関する変数全てが  $X_I$  に属するものとする. 以下, 本システムにおけるスケジュールリングアルゴリズムの各ステップについて述べる.

- 各エージェントは,  $X_i$  に対する初期値の割り当て  $A_i$  を作成し, リーダーエージェントにその評価値を送信する. このためリーダーエージェントは全体での評価値を計算することが出来る. また, 各  $P_I$  中に全  $X_I$  のリモートコピー  $X'_I$  を作成する.
- リーダーエージェントはローカル変数とリモートコピーからなる重み付き CSP に対する割り当てを更新し, 分散重み付き CSP 全体の評価値  $A$  を再計算する. この評価値と更新された  $X_I$  の情報を各エージェントに送信し, 次のエージェントにリーダーの権利を渡す. 各エージェントはこの手続きを繰り返す.
- 評価値の更新が一定の範囲に収まった時点で終了する.

本システムでは, 変数のリモートコピーを作成する際, greedy repair distributed method の場合に比べ, 分散 CSP 全体に対し, 各エージェントがより正確に把握することが出来る. 1 エージェントが持つ重み付き CSP の違いを, 図 2 における  $P_1$  に相当するエージェントのリモートコピー作成後の重み付き CSP ( $P_1'$ ) を例として図 3 に示す. 図 3(a) が greedy repair distributed method の場合の  $P_1'$  の例であり, 図 3(b) が本システムにおける  $P_1'$  の例である.



(a)  $X_I'$  = エージェント間変数 (b)  $X_I'$  = public変数全体とした場合のみの場合

図 3: 1 エージェントあたりの重み付き CSP

## 6 おわりに

分散するカレンダー情報の共有システムをマルチエージェントにより実装し, 論文投稿の際に行うミーティングの立案計画を対象に, 日程のスケジュールリングを可能にした. スケジュールリングの際, 各個人は実験等のために一定の時間を確保するものとし, スケジュールリングを分散重み付き制約充足問題として定式化した. この定式化により, ユーザ個人の要望をより反映できるスケジュールリングを実現した.

## 参考文献

- [1] Takuo Tsuruta, and Toramatsu Shintani "Scheduling Meetings Using Distributed Valued Constraint Satisfaction Algorithm," In the Proceedings of the Fourteenth European Conference on Artificial intelligence (ECAI2000), pp.283-387, Aug.2000.
- [2] M.Lemaitre, and G.Verfaillie, "An incomplete method for solving distributed valued constraint satisfaction," In Proc. of AAI-97 Workshop on Constraints and Agents, (1997)