ペトリネットの分解による AGV のタスク割当てと経路計画の同時最適化*

田中 友貴[†]・西 竜志[†]・乾口 雅弘[†]

Petri Net Decomposition Method for Simultaneous Optimization of Task Assignment and Routing for AGVs*

Yuki TANAKA[†], Tatsushi NISHI[†] and Masahiro INUIGUCHI[†]

We propose a simultaneous optimization method for task assignment and routing problems with multiple AGVs by the decomposition of Petri Nets. In the proposed method, Petri Net is decomposed into several subnets representing task subproblems and AGV subproblems. Each subproblem is solved by Dijkstra's algorithm. The subproblem on each subnet is repeatedly solved until a feasible solution for the original problem is derived. A solution method for subproblems with no final marking is newly developed. The effectiveness of the proposed method is demonstrated by comparing the performance with CPLEX as well as a nearest neighborhood heuristic method.

1. はじめに

半導体工場, コンテナターミナル, 生産システムなどにおける搬送システムでは, 複数台の自律無人搬送車 (AGV: Automated Guided Vehicle) が広く用いられており, AGV 間の衝突やデッドロックが生じることなく, 搬送時間を最短にする搬送経路を高速に作成することが求められている. AGV の経路計画手法に関しては従来から幅広く研究されており [1], シミュレーション技法 [2] やグラフ探索 [3], マルチエージェントネット [4], ペトリネットを用いたデッドロック回避による経路計画法 [5,6] などが提案されている. 従来の研究では, 搬送要求の AGV への割当てはあらかじめ定められているという条件, あるいは AGV の衝突やデッドロックを考慮しない条件下でのタスク割当てを採用した AGV 経路計画問題がおもに対象とされてきた [7]. しかしながら, 実際の搬送システムでは, リアルタイムに与えられる搬送要求に対して, AGV 間の干渉を考慮して AGV のタスク割当てと経路計画を同時に最適化する必要がある. この

ような動的問題は, ある時点で与えられたタスク集合に対して総搬送時間を最短にするような AGV へのタスク割当て, および経路計画を決定する静的問題を繰り返し解くことで解決できる. 本論文では, 上記の静的問題に対して, ペトリネットを用いた AGV のタスク割当てと経路計画の同時最適化手法を提案する.

ペトリネットは生産システム, 通信システム, 交通システムをはじめ, さまざまな離散事象システムを图的, かつ代数的に記述可能な汎用モデルである. ペトリネットの動作は, 初期マーキングから発火可能なトランジションの発火系列を定めることによって決定する. 簡単な構造をもつペトリネットに対してさえ, 初期マーキングから到達可能な目標マーキングまでの発火系列を決定する問題は NP 完全であることが知られている [8]. よって, 与えられた目的関数を最小化 (または最大化) する発火系列を求めるという最適発火系列問題は NP 困難となる. 最適発火系列問題の解法として, 一般にヒューリスティック探索 [9] が利用される. しかしながら, ペトリネットの規模が大きくなるにつれて, 状態空間爆発の問題が生じる. このような問題に対する有効な手法として, ペトリネットの分解と調整による最適化法が提案されている [10,11]. この手法は, ペトリネットを複数のサブネットに分解し, 各サブネットにおける部分問題の解の導出と調整を繰り返すことで全体として実行可能な準最適解を得る最適化手法である. Liao ら [12] は, ラグランジュ緩和により全体のペトリネットをタスクごとの部

* 原稿受付 2008 年 6 月 20 日

[†] 大阪大学 大学院 基礎工学研究科 システム創成専攻 社会システム数理領域 Division of Mathematical Science for Social Systems, Department of Systems Innovation, Graduate School of Engineering Science, Osaka University; 1-3 Machikaneyama-cho, Toyonaka city, Osaka 560-8531, JAPAN

Key Words: automated guided vehicle, task assignment and routing, Petri Net, decomposition, optimization.

分問題に分解し, AGV のスケジューリングを行う方法を提案している. ただし, AGV 間の干渉回避は各 AGV の経路計画問題において考慮されていない.

AGV のタスク割当てと経路計画の同時決定問題を対象として, 従来のペトリネット分解による最適化法を適用した場合, 目標マーキングの存在しないサブネットが生成される. このような分解を行った場合, 従来法では解が得られない. 本論文では, ペトリネットの分解による最適化手法において, 新たに目標マーキングのないサブネットに対する部分問題の解法を提案し, AGV のタスク割当てと経路計画の同時最適化へ適用する. そして, 提案手法の有効性を数値実験により検証する.

本論文は, 以下の章から構成される. 2. 章では, AGV のタスク割当てと経路計画の同時決定問題に対するペトリネットによるモデル化について述べる. 3. 章では, ペトリネットの分解, および最適化アルゴリズムについて述べる. 4. 章では, 数値実験結果を示す. 5. 章では, まとめと今後の課題を述べる.

2. ペトリネットによる AGV のタスク割当てと経路計画のモデリング

2.1 問題設定

最大 4 方向への移動, または停止が可能な地点を表す n 個のノードと, ノード間を接続した単一方向, あるいは双方向に移動可能な搬送路を表す同一長のエッジから構成される搬送システムを考える. 搬送システムには m 台の AGV が存在し, 時刻 0 において l 個のタスクが与えられるとする. ただし, 複数の AGV が同一時間において同一ノードに存在できない(衝突回避). また, 同一エッジを同時に占有できない(すれ違い回避). AGV の荷積み(荷下し)に要する時間は, AGV の隣接ノードへの移動に要する時間に対し, 十分長いと考えられる. したがって, AGV の隣接ノードへの移動に要する時間を 1 単位時間, 荷積み(荷下し)に要する時間を単位時間の正整数倍とする. 単位時間の正実数倍となるような場合には, 搬送システムのノードの間隔をより細かく設定し, 単位時間をより短く定めることで対応できる. 以下では, 説明を簡単にするため, 荷積み(荷下し)に要する時間は 1 単位時間とする. AGV の速度は一定かつ始動, 停止, 方向転換に要する時間は移動時間に比べて無視できると仮定する. 各タスクは 1 個の搬送先ノードを有する. 1 台の AGV に対して複数個のタスクを割り当てることが可能であり, タスクの割り当てられていない AGV が存在してもよい.

AGV のタスク割当てと経路計画の同時決定問題とは, 上記の搬送システムにおいて各 AGV の初期位置が与えられたとき, 各タスクの完了時間の総和(完了時間和)が最短となるような AGV へのタスク割当てと, 各 AGV の搬送経路を決定する問題である. 本論文では, 時間に関する最適化問題を対象としているため, 時間ペトリネッ

トモデルを用いるが, トランジション時間ペトリネット(TTPN)においてすべてのトランジションの発火継続時間が 1 のモデルはブレース/トランジションネット(PN)に一致するため, 以下では, すべてのトランジションの発火継続時間が 1 の時間ペトリネットを PN とよぶ.

2.2 ペトリネットによるモデリング

PN は $PN = (P, T, w, M_0)$ という四つの順序対で記述される. $P = \{p_1, p_2, \dots, p_{|P|}\}$ はブレースの有限集合, $T = \{t_1, t_2, \dots, t_{|T|}\}$ はトランジションの有限集合, $w: (P \times T) \cup (T \times P) \rightarrow \mathbb{Z}^+$ は P と T の接続関係, $M_0: P \rightarrow \mathbb{Z}^+$ は初期マーキングを表す. 時刻を $k \in \mathbb{Z}^+$ とすると, 発火ベクトル $r_k: T \rightarrow \{0, 1\}$ は, 時刻 k の T の発火の有無(発火する場合は 1, 発火しない場合は 0)を表し, マーキング $M_k: P \rightarrow \mathbb{Z}^+$ は, 時刻 k にブレース上に置かれているトークンの個数を表す. 二つの接続行列 A^+, A^- は, $(A^+)_{ji} = w(t_j, p_i)$, $(A^-)_{ji} = w(p_i, t_j)$ で定義され, 発火条件 $M_k - (A^-)^T r_k \geq 0$, 状態方程式 $M_{k+1} = M_k + (A^+ - A^-)^T r_k$ に基づき遷移する.

つぎに, 搬送システムのペトリネットによるモデリングについて述べる. AGV v_j ($1 \leq j \leq m$) がノード s_x ($1 \leq x \leq n$) に存在する状態を p_{v_j, s_x} , AGV v_j がノード s_a からノード s_b ($a \neq b$) へ移動する事象を t_{v_j, s_a, s_b} で表現する. Fig. 1 左図のような搬送経路を PN でモデル化した場合, Fig. 1 右図のようになる. ただし, 2.1 節の問題設定より, 隣接するノード間の移動時間は 1 単位時間とする. AGV の動作干渉を防ぐため, リソースブレース p_{o, s_x} を導入する. Fig. 1 左図において, AGV v_1 がノード s_1 からノード s_3 へ, AGV v_2 がノード s_4 からノード s_5 へ移動する場合, リソースブレースを導入した PN は, Fig. 2 になる. タスク u_i ($1 \leq i \leq l$) については, 割当て待ち状態を $p_{u_i, 0}$, AGV v_j にタスクを割り当てるという事象を $t_{u_i, v_j, 0}$, AGV v_j の到着待ち状態を p_{u_i, v_j} , AGV v_j の荷積み(荷下し)動作を $t_{u_i, v_j, 1}$, タスクの完了を $p_{u_i, 1}$ で表す. タスクの割当て, および荷積み(荷下し)に要する時間は 1 単位時間である. タスク u_i の搬送先ノードは, $t_{u_i, v_j, 1}$ が相互に接続している搬送経路上のブレースが示すノードである. Fig. 3 左図に示す搬送システムにおいて, 2 個のタスク u_1, u_2 (搬送先はそれぞれノード s_1, s_2) が与えられたとき, PN は Fig. 3 右図になる.

Fig. 1 PN modeling for AGVs system

Fig. 2 Introduction of resource place

Fig. 3 PN modeling for task assignment and routing

2.3 最適発火系列問題の定義と定式化

PN の最適発火系列問題とは, $PN = (P, T, w, M_0)$, 目標マーキング $M_f: P \rightarrow (\mathbb{Z}^+ \cup \{*\})$ (*はドントケア項), 全計画期間 $N_t \in \mathbb{N}$, 目的関数 J が与えられたとき, $M_{N_t} = M_f$ を満たす発火系列 $(r_0, r_1, \dots, r_{N_t-1}) \in (\{0, 1\})^{N_t}$ のうち, J を最小とするような発火系列を決定する問題である. タスク u_i の割当てを表すペトリネットのプレース集合を $P_{u_i} = \{p_{u_i,0}, p_{u_i,1}, p_{u_i,v_j}\} (1 \leq i \leq l; 1 \leq j \leq m)$ とする. 時刻 k のマーキングは M_k であり, $M_k(p)$ を $p \in P_{u_i}$ について並べた列ベクトルを $M_k^{u_i} \in \mathbb{Z}^{+P_{u_i}}$ とする. ここで, タスク u_i が完了していなければ 1, 完了後は 0 となる遷移関数 $\delta_{u_i,k} \in \{0, 1\}$ を定義すると, タスク u_i の完了時間は $\sum_{k=0}^{N_t} \delta_{u_i,k}$ で表される. タスクに対する総和

$$J = \sum_{i=1}^l \sum_{k=0}^{N_t} \delta_{u_i,k} \quad (1)$$

が完了時間であり, この最適化問題の目的関数となる. $M_f^{u_i}$ をタスク u_i についての目標マーキングとすると, $\delta_{u_i,k}$ を表す論理式は以下ようになる.

$$\delta_{u_i,k} = \begin{cases} 1 & (M_k^{u_i} \neq M_f^{u_i}) \wedge (\delta_{u_i,k-1} = 1) \\ 0 & (M_k^{u_i} = M_f^{u_i}) \vee (\delta_{u_i,k-1} = 0) \end{cases} \quad (2)$$

また, $M_f^{u_i}$ はタスク u_i の完了を表すプレース $p_{u_i,1}$ のみ 1, それ以外はドントケア項(*)であるから, 次の関係式が成り立つ.

$$(M_f^{u_i})^T M_k^{u_i} = \begin{cases} 1 & (M_k^{u_i} \neq M_f^{u_i}) \\ 0 & (M_k^{u_i} = M_f^{u_i}) \end{cases} \quad (3)$$

したがって, タスク l 個, AGV m 台におけるタスク割当てと経路計画の同時最適化問題は, 以下の最適発火系列問題として定式化される.

$$\min_{r_k} J = \sum_{i=1}^l \sum_{k=0}^{N_t} \delta_{u_i,k} \quad (4)$$

subject to (2), (3)

$$M_k - (A^-)^T r_k \geq 0 \quad (5)$$

$$M_{k+1} = M_k + (A^+ - A^-)^T r_k \quad (6)$$

$$\delta_{u_i,k} \in \{0, 1\} \quad (7)$$

3. ペトリネットの分解と調整による最適化

3.1 ペトリネットの分解

タスク数や AGV 台数の増加に伴い, 最適化問題を表すペトリネットは大規模化するため, 最適発火系列問題を解くことは困難になる. そこで, ペトリネットを複数のサブネットに分解し, サブネットに関する部分問題を繰り返し最適化することにより, 全体の問題を最適化するという方法をとる. 対象とする最適発火系列問題が以下の条件を満たしているとき, ペトリネットの分解が可能である.

- 1) 目的関数 J は, 個々の要素 $u_i (1 \leq i \leq M)$ についての目的関数 J_{u_i} の和 $J = \sum_{i=1}^M J_{u_i}$ として表現される.
- 2) 複製プレース集合 P_R に関して, 目標マーキングが定義されていない ($\forall p \in P_R$ に対する目標マーキングはドントケア項(*)である).

タスク l 個, AGV m 台におけるタスク割当てと経路計画の同時決定問題は, タスク u_i の搬送先ノードを表すプレースと, 衝突回避のために導入したリソースプレースを複製することで, タスク割当てに関する l 個のサブネットと AGV の経路計画に関する m 個のサブネットに分解される. たとえば, Fig. 3 右図に示す PN をサブネットに分解した場合, Fig. 4 に示す四つのサブネットが生成される.

以降, 要素 $u_i (1 \leq i \leq M)$ は, $1 \leq i \leq l$ のときタスク u_i を表し, $(l+1) \leq i \leq M$ のとき AGV v_j を表すとする (ただし, $M = l + m$). トランジション集合 T は, (8) 式に示す M 個の部分集合 T_{u_i} に分解される.

$$T = \bigcup_{i=1}^M T_{u_i} \quad (i \neq j, T_{u_i} \cap T_{u_j} = \emptyset) \quad (8)$$

また, プレース集合 P の部分集合 P_{u_i} , 複製プレース集合 P_R は以下のように定義される. ここで, $\bullet p$ は p に対する入力トランジション集合, $p\bullet$ は p に対する出力トランジション集合である.

Fig. 4 Decomposition of PN into subnets

$$P_{u_i} = \{p \mid p^\bullet \subseteq T_{u_i}, \bullet p \subseteq T_{u_i}\} \quad (9)$$

$$P_R = P \setminus \bigcup_{i=1}^{\mathcal{M}} P_{u_i} \quad (i \neq j, P_{u_i} \cap P_{u_j} = \emptyset) \quad (10)$$

ここで, $r_k(t)$ を $t \in T_{u_i}$ について, $M_k(p)$ を $p \in P_{u_i}$, および $p \in P_R$ について並べた列ベクトルをそれぞれ $r_k^{u_i} \in \{0,1\}^{T_{u_i}}$, $M_k^{u_i} \in \mathbb{Z}^{+P_{u_i}}$, $M_k^R \in \mathbb{Z}^{+P_R}$ とすると, 適当な整数行列 $A_{u_i}^+ \in \mathbb{Z}^{+T_{u_i} \times P_{u_i}}$, $A_{u_i}^- \in \mathbb{Z}^{+T_{u_i} \times P_{u_i}}$, および

$$B^+ = [(B_{u_1}^+)^T, \dots, (B_{u_{\mathcal{M}}}^+)^T]^T \quad (B_{u_i}^+ \in \mathbb{Z}^{+T_{u_i} \times P_R})$$

$$B^- = [(B_{u_1}^-)^T, \dots, (B_{u_{\mathcal{M}}}^-)^T]^T \quad (B_{u_i}^- \in \mathbb{Z}^{+T_{u_i} \times P_R})$$

を用いて, PN の発火条件を

$$M_k^{u_i} - (A_{u_i}^-)^T r_k^{u_i} \geq 0 \quad (1 \leq i \leq \mathcal{M}) \quad (11)$$

$$M_k^R - (B^-)^T r_k \geq 0 \quad (12)$$

状態方程式を以下のように表すことができる.

$$M_{k+1}^{u_i} = M_k^{u_i} + (A_{u_i}^+ - A_{u_i}^-)^T r_k^{u_i} \quad (13)$$

$$M_{k+1}^R = M_k^R + (B^+ - B^-)^T r_k \quad (14)$$

以上に示したブレース, トランジション集合の分解により, PN は要素 u_i ごとのサブネット $PN^{u_i} = (P_{u_i} \cup P_{R_{u_i}}, T_{u_i}, w^{u_i}, M_0^{u_i})$ に分解される. ただし, $P_{R_{u_i}}$ は P_R と一対一の関係をもつブレース集合である. ペトリネットの分解条件から, PN における目標マーキングはブレース集合 P_R に関してドントケア項 (*) であるため, PN^{u_i} における目標マーキングは以下のように表される.

$$M_f^{u_i}(p) = M_f(p) \quad (\forall p \in P_{u_i}) \quad (15)$$

PN^{u_i} のマーキングを $p \in P_{R_{u_i}}$ について並べた列ベクトルを $M_k^{R_{u_i}} \in \mathbb{Z}^{+P_{R_{u_i}}}$ とすると, PN^{u_i} における発火条件と状態方程式は (11), (13) 式, および以下の式で表さ

れる.

$$M_k^{R_{u_i}} - (B_{u_i}^-)^T r_k^{u_i} \geq 0 \quad (16)$$

$$M_{k+1}^{R_{u_i}} = M_k^{R_{u_i}} + (B_{u_i}^+ - B_{u_i}^-)^T r_k^{u_i} \quad (17)$$

つぎに, 要素 u_i の目的関数について述べる. タスク u_i の割当てを表すサブネットには $M_f^{u_i}(p) \in \mathbb{Z}^+$ なる $p \in P_{u_i}$ が存在するため, (2) 式を用いて表されるタスク u_i の完了時間が目的関数となる. しかしながら, AGV v_j に関するサブネットには目標マーキングが存在しないため, (2) 式により目的関数を定義できず, このままでは AGV が不必要な移動を行う可能性がある. そこで, AGV v_j の移動距離を目的関数として与えるため, 遷移関数を以下のように定義する.

$$\delta'_{u_i,k} = \begin{cases} 1 & (M_k^{u_i} \neq M_{k-1}^{u_i}) \\ 0 & (M_k^{u_i} = M_{k-1}^{u_i}) \end{cases} \quad (M_f^{u_i} \in \{*\}^{P_{u_i}}) \quad (18)$$

(18) 式より, AGV v_j の移動距離に比例する移動時間は $\sum_{k=0}^{N_t} \delta'_{u_i,k}$ と表すことができる. したがって, 要素 u_i の目的関数は次式で表される.

$$J_{u_i} = \sum_{k=0}^{N_t} \delta_{u_i,k} \quad (19)$$

$$\delta_{u_i,k} = \begin{cases} (2) & (1 \leq i \leq l) \\ (18) & (l+1 \leq i \leq \mathcal{M}) \end{cases} \quad (20)$$

3.2 最適化アルゴリズム

各サブネットに対して独立に求めた発火系列は, もとのペトリネットで実行可能とは限らない. もとのペトリネットで実行可能な発火系列を得るため, (1) 式の目的関数に発火条件違反に対するペナルティを導入する. ペナルティ法を用いた最適化アルゴリズム [11] を以下に示す.

Step 1 初期最適化

繰返し回数 $N := 1$ とし, 要素 u_j ($j = 1, 2, \dots, \mathcal{M}$) について, (21) 式の部分問題に対する最適解を 3.3 節の解法で求め, 暫定解 $\hat{r}_k^{u_j} := r_k^{u_j}$ とする.

$$\min_{\{r_k^{u_j}\}} J_{u_j} \quad (21)$$

subject to (11), (13), (16), (17)

Step 2 収束判定

暫定解 $\hat{r}_k = [(\hat{r}_k^{u_1})^T, (\hat{r}_k^{u_2})^T, \dots, (\hat{r}_k^{u_{\mathcal{M}}})^T]^T$ が (12) 式を満たすかどうか (実行可能かどうか) を判定する. 実行可能かつ解の更新が行われていなければ, 暫定解を準最適解として出力し, アルゴリズムを終了する.

Step 3 再度最適化

再度最適化を行う要素を選ぶ. 前回の再度最適化時に u_i ($i \neq \mathcal{M}$) が選ばれていたのであれば u_{i+1} を選択し, $u_{\mathcal{M}}$ が選ばれていたのであれば u_1 を選択する. 再度最適化を行う要素 u_j について, u_j 以外の暫定解 $\hat{r}_k^{u_i}$ ($u_i \neq u_j$) を定数として固定し, (22) 式の部分問題を 3.3 節の解法

で解く．ここで， $\omega_{p,u_j}^{(N)}$ は N 回目の再度最適化におけるペナルティの重み係数である．

$$\min_{\{r_k^{u_j}\}} \left(J_{u_j} + \sum_{k=0}^{N_t} \sum_{p \in P_{Ru_j}} \omega_{p,u_j}^{(N)} \alpha_{u_j,p,k}(r_k^{u_j}) \right) \quad (22)$$

subject to (11), (13), (16), (17), (23)–(27)

ペナルティ関数 $\alpha_{u_j,p,k}(r_k^{u_j})$ は， $r_k^{u_j}$ により時刻 k にプレース $p \in P_R$ についての発火条件が満たされないとき，実行可能となるまでに必要なトークン数を表す．前回の再度最適化によって得られる暫定解に対するマーキングを $\hat{M}_k$ ，発火列を $\hat{r}_k$ とすると， $\alpha_{u_j,p,k}(r_k^{u_j})$ は (23)–(27) 式で表すことができる [11]．ここで， $B_{u_j}^-(p)$ は行列 $(B_{u_j}^-)^T$ 内のプレース p に対応する行ベクトルである． $K_{u_j,k}(p)$ は列ベクトル $K_{u_j,k}$ のプレース p に対応する要素であり，定数となる．

$$\alpha_{u_j,p,k}(r_k^{u_j}) = \max\{0, -\alpha'_{u_j,p,k}\} \quad (23)$$

$$\alpha'_{u_j,p,k} = K_{u_j,k}(p) + M_k^{R_{u_j}}(p) - B_{u_j}^-(p)r_k^{u_j} \quad (24)$$

$$K_{u_j,k} = (\hat{M}_k^R)^- - (B_{u_j}^+ - B_{u_j}^-)^T \hat{\Sigma}_{k-1}^{u_j} + (B_{u_j}^-)^T \hat{r}_k^{u_j} - M_0^{R_{u_j}} \quad (25)$$

$$(\hat{M}_k^R)^- = \hat{M}_k^R - (B^-)^T \hat{r}_k \quad (26)$$

$$\hat{\Sigma}_{k-1}^{u_j} = \hat{r}_0^{u_j} + \hat{r}_1^{u_j} + \dots + \hat{r}_{k-1}^{u_j} \quad (27)$$

$$(p \in P_{Ru_j}; k = 0, 1, \dots, N_t)$$

Step 4 ペナルティの重み更新

要素 u_j について，Step 3 で得られた解 $r_k^{u_j}$ と暫定解 $\hat{r}_k^{u_j}$ ($i \neq j$) を用いて $\alpha_{u_j,p,k}(r_k^{u_j})$ を計算し，(28), (29) 式を用いてペナルティの重み係数の更新を行う． $\Delta\omega$ はペナルティ項に対する重み係数の更新幅である．

$$\omega_{p,u_j}^{(N+1)} = \omega_{p,u_j}^{(N)} + \Delta\omega \sum_{k=0}^{N_t} \alpha_{u_j,p,k}(r_k^{u_j}) \quad (28)$$

$$\omega_{p,u_i}^{(N+1)} = \omega_{p,u_i}^{(N)} \quad (i \neq j) \quad (29)$$

Step 5 暫定解の更新

Step 3 で部分問題を解くことによって得られた解 $r_k^{u_j}$ を暫定解とし， $\hat{r}_k^{u_j} := r_k^{u_j}$ とする． $N := N + 1$ として Step 2 へ戻る．

3.3 部分問題に対する解法

(21), (22) 式で表される部分問題は整数計画問題として定式化される．タスク割当てと経路計画の同時決定問題をタスク割当てに関するサブネットと各 AGV の経路計画に関するサブネットに分解したとき，目標マーキングをもたないサブネットが生成される．しかし，各サブネットの総トークン数は一つとなり，分解されたサブネットによって生成される状態数がサブネットの大きさに対して多項式オーダの範囲となるため，サブネットのサイズが増加したとしても，ダイクストラ法を用いて部分問題を効率的に解くことができる．タスク割当てに関

するサブネットは目標マーキングが存在するため， J_{u_j} を列ベクトル $M_k^{u_j}$ が初期マーキング $M_0^{u_j}$ から目標マーキング $M_f^{u_j}$ に到達するまでの時間とし，状態集合 $\mathcal{P}_s^j$ は以下のように定義できる． h_k は時刻 k に目標マーキングに到達したかどうかを表す 0–1 変数 (到達した場合 1，到達していない場合 0) である．

$$\mathcal{P}_s^j = \{\sigma \mid \sigma = (M_k^{u_j}, k, h_k); 0 \leq k \leq N_t; h_k \in \{0, 1\}; h_k = 1 \iff M_k^{u_j} = M_f^{u_j}\} \quad (30)$$

時刻 k において $M_k^{u_j}$ が目標マーキング $M_f^{u_j}$ と一致したとき， $(M_k^{u_j}, k, 1)$ の状態が生成される． $(M_k^{u_j}, k, 1)$ は，タスク完了の状態であり，その後は $M_k^{u_j} = M_f^{u_j}$ なるマーキング以外に遷移できない．ある状態 $\sigma_a = (M_k^{u_j}, k, h_k) \in \mathcal{P}_s^j$ から状態 $\sigma_b = (M_{k+1}^{u_j}, k+1, h_{k+1}) \in \mathcal{P}_s^j$ への遷移に対する時間コストは， $h_k = 0$ のとき 1 となり， $h_k = 1$ のとき 0 となる．状態 σ_a から状態 σ_b への遷移コストは，時間コストとペナルティコストの和として以下のように表される．

$$d_{u_j,\sigma_a,\sigma_b} = \delta_{u_j,k} + \sum_{p \in P_{Ru_j}} \omega_{p,u_j}^{(N)} \alpha_{u_j,p,k}(r_k^{u_j}) \quad (31)$$

AGV の移動を表すサブネットは J_{u_i} を (18) 式で定義される距離に比例する遷移時間ととし，状態集合を以下のように定義する．

$$\mathcal{P}_s^j = \{\sigma \mid \sigma = (M_k^{u_j}, k); 0 \leq k \leq N_t\} \quad (32)$$

なお，状態遷移コストは同様に (31) 式で表される．目標マーキングをもたないサブネットに対しては，コスト関数を最小とする発火系列を求める問題となる．状態集合 $\mathcal{P}_s^j$ をノード集合， $d_{u_j,\sigma_a,\sigma_b}$ を σ_a から σ_b への経路長と置き換えれば，ダイクストラ法を用いて (21), (22) 式的最適解を求めることができる．計算量は，ノード数 n に対して $O(n^2)$ となる．

3.4 目標マーキングのないサブネットに対する部分問題の解法

Fig. 5 のようにタスク数 1，AGV 台数 1 の搬送システムを表すペトリネットをサブネットに分解した場合を考える．タスクに関するサブネットには目標マーキングが存在するが，AGV に関するサブネットには目標マーキングが存在しない．そのため，AGV に関するサブネットでは，AGV の移動距離と発火条件違反によるペナルティの和を最小とするような発火系列を決定する．Fig. 6,

Fig. 5 Subnets for 1 task and 1 AGV system

Fig. 7 に AGV に関するサブネットの可到達状態グラフを示す．初期最適化により，タスクに関するサブネットは $k=1$ で AGV がノード 1 に存在するような解を AGV に関するサブネットに対して要求する．しかし，AGV に関するサブネットにおいてそのような解は存在しない．(23) 式のペナルティ関数の場合，Fig. 6 に示されるような可到達状態グラフが作成される．図中の各ノード内の番号は Fig. 5 における AGV の位置を表すノード番号，枝に付けられた数は遷移コストを表す．

Fig. 6 Reachability state graph for conventional penalty function

Fig. 7 Reachability state graph for modified penalty function

従来のペナルティ関数では，最小コストとなる解は AGV がノード 4 から移動しない発火系列であるから，AGV に関するサブネットの解は初期解から変化しない．したがって，3.2 節の最適化アルゴリズムにおいて繰返しによる解の調整がうまく機能しない．そこで，目標マーキングのないサブネットに対して，以下のペナルティ関数を採用する．

$$\alpha_{u_j,p,k}(r_k^{u_j}) = \max\{0, -\alpha'_{u_j,p,k} + \alpha_{u_j,p,k-1}\} \quad (33)$$

(33) 式のペナルティ関数は，トークン不足数 $(-\alpha_{u_j,p,k})$ と時刻 $k-1$ でのペナルティ $\alpha_{u_j,p,k-1}$ により構成される．つまり，時刻 $k-1$ で生じたペナルティが，トークン不足数として時刻 k のペナルティに持ち越される．よって，(33) 式のペナルティ関数の場合，Fig. 7 に示す可到達状態グラフが作成され，最小コストとなる解は $k=2$ に AGV がノード 1 に存在する発火系列となる．AGV に関するサブネットの解の変更により，タスクに関するサブネットも $k=2$ で AGV がノード 1 に存在するような

解を AGV に関するサブネットに対して要求するように変更される．このようにして，全体として実行可能な解が 3.2 節の最適化アルゴリズムにより求められる．

4. 数値実験

本節では，提案法と汎用ソルバ(CPLEX)による厳密解法，および最近傍法(nearest neighbor method)によるタスク割当てを用いた最適化法との比較を行う．使用した搬送経路は，Fig. 8 に示す 115 ノード・128 辺の単一方向経路である．AGV 台数 5 台，タスク数 10 個の場合， $|P|=760$, $|T|=740$ となる．ペナルティの重み更新幅は $\Delta\omega=0.3$ とした．使用した計算機は，Pentium D 3.4GHz，メモリ 1GB である．

Fig. 8 Unidirectional transportation system with 115 nodes and 128 edges

4.1 汎用ソルバとの比較

汎用ソルバとして ILOG 社の CPLEX10.1 を使用し，2.3 節で定式化した数理計画モデルに対する厳密解を求める．AGV 5 台，タスク 10 個において 10 ケースの数値実験を行った結果を Table 1 に示す．各ケースにおいて，AGV の初期位置およびタスクの搬送先ノードはランダムに与えられる．また，Table 1 中の TCT はタスク完了時間の総和である．Table 1 より，提案手法では平均して 47.3 秒の計算時間で解を導出できるが，CPLEX では最適解を導出するために，平均して約 12,000 秒以上の計算時間を必要とした．提案法はすべてのケースにおい

Table 1 Comparison of performance between the proposed method and CPLEX

Case	l	m	Proposed method		CPLEX	
			Time[s]	TCT	Time[s]	TCT
1	10	5	54.4	171	4738.3	171
2	10	5	32.1	107	100.5	105
3	10	5	48.2	139	154.9	139
4	10	5	69.5	208	52012.0	189
5	10	5	26.2	85	79.6	79
6	10	5	59.2	187	64649.4	187
7	10	5	50.6	161	2244.0	151
8	10	5	42.5	161	1040.1	159
9	10	5	47.7	98	336.5	98
10	10	5	42.2	145	1255.5	142
Average			47.3	146.2	12661.1	142

て CPLEX よりも短時間で実行可能解が得られていることがわかる．また，4 ケース (Case 1, 3, 6, 9) においては，提案法により最適解が得られている．以上より，提案法は厳密解法より短時間で最適に近い解が得られることを確認した．

4.2 ヒューリスティックルールによるタスク割当てとの比較

本節では，提案手法とタスク割当てに最近傍法，経路計画にペトリネットの分解による最適化法 [11] を採用した手法との比較を行うことにより，同時最適化の効果を検証する．最近傍法によるタスク割当てでは，搬送時間の下界値が最小となるような AGV にタスクが割り当てられる．AGV v_j の下界値 E_{v_j} は次式で求められる [7]．

$$E_{v_j} = \max\{\tau_{free}^{v_j}, \tau_{now}\} - \tau_{now} + \tau_{min}^{v_j} \quad (34)$$

$\tau_{free}^{v_j}$ は AGV v_j が所有している全タスクの処理を完了する予定時刻， τ_{now} は現在の時刻， $\tau_{min}^{v_j}$ は AGV v_j が所有しているタスクのうち，最後に処理するタスクの搬送先ノードから新たに発生したタスクの搬送先ノードまでの最短搬送時間を表す．2.1 節で述べた問題設定より，時刻 0 でのタスク割当てを想定しているため， $\tau_{now} = 0$ である．AGV が 5 台，10 台，15 台，タスクが 5 個，10 個，15 個，20 個という問題設定で，各 5 ケースの数値実験を行った結果を Table 2 に示す．Table 2 中の l はタスク数を， m は AGV 台数を表す．Table 2 より，ほぼすべてのケースにおいて，提案法によって得られた完了時間と最近傍法よりも良い値である．タスク数が少ない場合は両者の差は小さく，AGV 台数よりもタスク数が多くなるにつれて両者の差は大きくなっていることがわかる．この原因は，最近傍法によるタスク割当てが他の AGV との干渉を考慮しないためであると考えられる．つまり，タスク数が少ない場合はフリーな AGV が多く存在し，搬送中の AGV 同士の干渉も少ないため，(34) 式によって得られる下界値は最適値に近いが，AGV 台数よりもタスク数が多くなると，搬送中の AGV 同士の干渉が増加し，(34) 式によって得られる下界値は最適値と大きく離れる．とくに，AGV 5 台，タスク 15 個や AGV 10 台，タスク 20 個の問題の場合，複数のタスクが 1 台の AGV に割り当てられるため，(34) 式による影響が顕著に表れている．これは，タスク割当てと経路計画の同時最適化は AGV 間の干渉を考慮する場合に効果が大きいことを意味する．

つぎに，問題の規模と計算時間について考察する．提案法の計算時間は，部分問題の平均計算時間と実行可能解を得るまでの繰返し回数の積に依存する．タスクに関するサブネットの部分問題で生成される状態数は，AGV に関するサブネットの部分問題で生成される状態数よりも少ないため，部分問題の平均計算時間については AGV 台数が大きく影響する．また，サブネット数 (タスク数

Table 2 Comparison of performance between the proposed method and nearest neighbor heuristic method

Case	l	m	Proposed method		NN method	
			Time[s]	TCT	Time[s]	TCT
1	5	5	23.8	55	1.9	55
2	5	5	27.1	52	2.5	52
3	5	5	27.8	61	2.1	61
4	5	5	31.5	97	3.2	97
5	5	5	7.1	21	0.6	21
6	10	5	53.9	171	14.1	201
7	10	5	31.7	107	4.3	112
8	10	5	47.6	139	5.7	165
9	10	5	68.7	208	8.2	197
10	10	5	25.9	85	1.8	79
11	15	5	70.6	226	16.6	266
12	15	5	65.9	234	15.5	278
13	15	5	47.5	167	27.8	169
14	15	5	85.5	318	15.7	344
15	15	5	53.7	232	6.8	240
16	15	10	147.5	122	8.1	132
17	15	10	162.5	136	15.9	146
18	15	10	132.2	111	5.4	113
19	15	10	143.4	102	8.7	106
20	15	10	159.2	144	12.5	158
21	20	10	179.7	156	32.8	183
22	20	10	209.2	171	14.8	183
23	20	10	167.6	159	20.1	169
24	20	10	164.7	144	15.6	169
25	20	10	247.4	217	73.1	269
26	20	15	394.6	114	51.9	124
27	20	15	611.4	156	24.1	160
28	20	15	498.2	150	38.7	165
29	20	15	437.6	135	24.2	145
30	20	15	421.5	130	25.0	143

と AGV 台数の和) が増加すると解の調整が困難となるため，繰返し回数についてはサブネット数が大きく影響する．したがって，サブネット数が多くなるほど，とくに AGV 台数が増加するほど提案法の計算時間は増加するといえる．実際，Table 2 より，サブネット数が多くなるほど平均計算時間が増加し，サブネット数が同じ場合には AGV 台数が多いほど平均計算時間が大きいことがわかる．提案法の計算時間は，最近傍法に劣るものの，すべてのケースで 620 秒以内に解が得られている．以上より，最近傍法によるタスク割当てを用いた場合，問題サイズが大きくなると最適解が得られないことを示した．また，提案法はタスク数が多いケースの問題に対して有効であることが数値実験によって確認された．

5. おわりに

本論文では，ペトリネットの分解による AGV のタスク割当てと経路計画の同時最適化手法を提案した．ペト

リネットの分解により生じた目標マーキングのないサブネットに対する部分問題の解法を示した．数値実験により，提案法はCPLEXによる厳密解法より短時間で，最適に近い解が得られることを示した．また，最近傍法によるタスク割当てを用いた最適化手法との比較を行い，提案法の有効性をタスク数20，AGV15台の規模の問題に対して確認した．今後の課題として，ヒューリスティックルールを用いた暫定解の実行可能化による提案法の計算時間の短縮や，ラグランジュ緩和による解の精度向上，および解の最適性評価が挙げられる．また，実際の搬送システムは搬送要求がリアルタイムで与えられるため，動的な搬送モデルに対して提案法を適用し，提案法の性能評価を行う必要がある．

参考文献

- [1] I. F. A. Vis: Survey of research in the design and control of automated guided vehicle systems; *Eur. J. Oper. Res.*, Vol. 170, No. 3, pp. 677–709 (2006)
- [2] S. Tunali: Evaluation of alternate routing policies in scheduling a job-shop type FMS; *Comp. Ind. Eng.*, Vol. 32, No. 2, pp. 243–250 (1997)
- [3] C. W. Kim and J. M. A. Tanchoco: Conflict-free shortest-time bidirectional AGV routing; *Int. J. Prod. Res.*, Vol. 29, No. 12, pp. 2377–2391 (1991)
- [4] 平石：ペトリネットによるマルチエージェントシステムのモデル化；システム/制御/情報，Vol. 45, No. 8, pp. 439–444 (2001)
- [5] M. Dotoli and M. P. Fanti: Coloured timed Petri net model for real-time control of automated guided vehicle systems; *Int. J. Prod. Res.*, Vol. 42, No. 9, pp. 1787–1814 (2004)
- [6] N. Q. Wu and M. Zhou: Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking; *IEEE/ASME Transactions on Mechatronics*, Vol. 12, No. 1, pp. 63–72 (2007)
- [7] 安藤，西，小西：分散並列型経路計画システムを用いた動的搬送問題に対する複数AGVの局所的再経路計画法；システム制御情報学会論文誌，Vol. 17, No. 4, pp. 162–169 (2004)
- [8] T. Watanabe, Y. Mizobata and K. Onaga: Time complexity of legal firing sequence and related problems of Petri Nets; *The Transactions of the IEICE*, Vol. E 72, No. 12, pp. 1400–1409 (1989)
- [9] D. Y. Lee and F. DiCesare: Scheduling flexible manufacturing systems using Petri nets and heuristic search; *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 2, pp. 123–132 (1994)
- [10] D. W. He, B. Strege, H. Tolle and A. Kusiak: Decomposition in automatic generation of Petri nets for manufacturing system control and engineering; *Int. J. Prod. Res.*, Vol. 38, No. 6, pp. 1437–1457 (2000)
- [11] 前野，西，小西：ペトリネットを用いたAGV経路計画問題の分解法による最適化；システム制御情報学会論文誌，Vol. 19, No. 11, pp. 426–434 (2006)
- [12] D. Y. Liao, M. Jeng and M. C. Zhou: Petri Net modeling and Lagrangian relaxation approach to vehicle scheduling in 300mm semiconductor manufacturing; *Proc. Int. Conf. Robotics and Automation*, pp. 5301–5306 (2004)

著者略歴

たなか ゆうき
田中 友貴

2008年3月大阪大学基礎工学部システム科学科卒業，同年大阪大学大学院基礎工学研究科博士前期課程システム創成専攻に入学し，現在に至る．離散事象システムの最適化に関する研究に従事．

にし たつし
西 電志（正会員）

2001年3月京都大学大学院工学研究科博士後期課程修了．同年4月岡山大学工学部助手，2006年3月大阪大学大学院基礎工学研究科助教授，2007年4月准教授となり，現在に至る．離散システムの最適化，スケジューリング，サプライチェーンの最適化に関する研究に従事．博士（工学）．スケジューリング学会，計測自動制御学会，日本OR学会，電気学会，日本機械学会，IEEEなどの会員．

いぬい ぐち まさひろ
乾口 雅弘（正会員）

1987年大阪府立大学大学院工学研究科博士前期課程修了．同年大阪府立大学助手，1992年広島大学助教授，1997年大阪大学助教授，2003年同教授，現在に至る．主として，ファジィ数値計画法，可能性理論およびラフ集合理論の意思決定問題への応用に関する研究に従事．博士（工学）．1997年日本ファジィ学会論文賞などを受賞．IEEE，計測自動制御学会，日本知能情報ファジィ学会，日本OR学会，電子情報通信学会などの会員．