The University of Osaka
Institutional Knowledge Archive

Title	サプライチェーンにおけるモデル化と意思決定の自動化に関する研究
Author(s)	岡, 宏憲
Citation	大阪大学, 2009, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2592
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

サプライチェーンにおけるモデル化と意思決定の自動化に関する研究

2009年3月

岡　宏憲

サプライチェーンにおけるモデル化と意思決定の自動化に関する研究

提出先　大阪大学大学院情報科学研究科

提出年月　2009年3月

岡　宏憲

研究業績目録

A. 学術論文誌論文

(1) 岡宏憲, 圓田哲也, 一階良知, 薦田憲久: "サプライチェーンにおける機能別モデル定義方式とそれに基づくシミュレーションツールの開発", 電気学会論文誌C（電子・情報・システム部門誌）, Vol. 123, No. 10, pp. 1884–1891 (2003.10).

(2) 岡宏憲, 長谷川努, 一階良知, 薦田憲久: "二者間交換分配交渉における目的分析知識を用いた交渉方式", 電気学会論文誌C（電子・情報・システム部門誌）, Vol. 124, No. 1, pp. 189–195 (2004.1).

(3) Yoshitomo Ikkai, Hironori Oka, and Norihisa Komoda: "An Integrated Tabu Search with Probabilistic Rule Application for Large-Scale Job Shop Scheduling Problems with Group Constraints", *WSEAS Transactions on Information Science & Applications*, Issue 12, Vol. 2, pp. 2031–2037 (2005.12).

B. 国際会議

(1) Tsutomu Hasegawa, Hironori Oka, Yoshitomo Ikkai, and Norihisa Komoda: "A Negotiation Method Using Improvement Knowledge for Bilateral Exchange and Distribution Negotiation", in *Proc. of 2001 IEEE International Conference on Systems, Man and Cybernetics (SMC'2001)* (in CD-ROM) (2001.10).

(2) Hironori Oka, Tsutomu Hasegawa, Yoshitomo Ikkai, and Norihisa Komoda: "A Bilateral Negotiation Method Using Satisfying Level with

Knowledge of Purpose Analysis", in *Proc. of 2001 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2001)*, Vol. 2, pp. 359–364 (2001.10).

(3) Hironori Oka, Tomoyuki Kitagawa, Yoshitomo Ikkai, and Norihisa Komoda: "An Application of a Simulation Tool Using a Multi-functional Modeling Method to Logistics Field", in *Proc. of IEEE International Symposium on Industrial Electronics 2002 (IEEE-ISIE2002)*, Vol. 4, pp. 1370–1375 (2002.7).

(4) Norihisa Komoda, Yoshitomo Ikkai, and Hironori Oka: "A Flexible Management Method of Supply Chain by Using Logistics Information with RF-ID Tag and Bluetooth", in *Abstracts of 16th JISR-IIASA Workshop on Methodologies and Tools for Complex System Modeling and Integrated Policy Assessment (CSM2002)*, pp. 24–25 (2002.7).

(5) Hironori Oka, Yoshitomo Ikkai, and Norihisa Komoda: "A Procurement Negotiation Method Using Objective Relaxation Knowledge for Appeasement", in *Proc. of 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA03)*, Vol. 2, pp. 245–250 (2003.9).

(6) Yoshitomo Ikkai, Hironori Oka, Norihisa Komoda, and Rei Itsuki: "An Autonomous Distributed Information System for Logistics Control with Data Carriers", in *Proc. of IEEE International Conference on Industrial Technology 2003 (ICIT 2003)*, pp. 1079–1082 (2003.12).

(7) Kentaro Tuchida, Masaki Tabuchi, Hironori Oka, Yoshitomo Ikkai, and Norihisa Komoda: "A Robust Scheduling Method for a Job Shop Problem in Production by Using Data Carriers", in *Proc. of 2004 IEEE International Conference on Systems, Man and Cybernetics (SMC2004)* (in CD-ROM) (2004.10).

(8) Yoshitomo Ikkai, Hironori Oka, and Norihisa Komoda: "A High-Speed Integrated Scheduling System with Tabu Search for Large-Scale Job

Shops Problems with Group Constraints", in *Proc. of 5th WSEAS International Conference on Systems Theory and Scientific Computation*, pp. 221–226 (2005.9).

(9) Masanori Akiyoshi, Mariko Matsumoto, Hironori Oka, and Norihisa Komoda: "Negotiation Protocol for Bilateral Semi-Competitive Trading Environments", in *Proc. of 10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2005)*, Vol. I, pp. 1101–1106 (2005.9).

(10) Hiroaki Chujo, Hironori Oka, Yoshitomo Ikkai, and Norihisa Komoda: "A Real-Time Production Scheduling Method Using Attractor Selection", in *Proc. of 5th International Conference on Computational Intelligence for Modeling Control and Automation (CIMCA'2005)*, Vol. 1, pp. 511–516 (2005.11).

(11) Hiroaki Chujo, Hironori Oka, Yoshitomo Ikkai, and Norihisa Komoda: "A Consideration of Application of Attractor Selection to a Real-Time Production Scheduling", in *Proc. of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pp. 346–354 (2006.1).

C. 学会講演

(1) 岡宏憲, 長谷川努, 一階良知, 薦田憲久: "二者間交換分配交渉における目的分析知識を用いた交渉方式", 電気学会情報システム研究会, IS-01-19, pp. 17–22 (2001.9).

(2) 岡宏憲, 一階良知, 薦田憲久: "機能別モデリングによるサプライチェーンシミュレーション方式の流通事例への適用", 電気学会情報システム研究会, IS-02-8, pp. 7–12 (2002.6).

(3) 岡宏憲, 圓田哲也, 一階良知, 薦田憲久: "サプライチェーンにおける機能別モデル定義方式とシミュレーションツールの開発", 平成 14 年電気学

会電子情報システム部門大会, OS-1-1 (in CD-ROM) (2002.9).

(4) 岡宏憲, 一階良知, 薦田憲久: “受発注業務における譲歩知識による提案改善型交渉方式”, 電気学会情報システム研究会, IS-03-3, pp. 7–12 (2003.3).

(5) 岡宏憲, 一階良知, 薦田憲久: “譲歩知識による提案改善型交渉方式の受発注業務事例への適用”, 平成15年電気関係学会関西支部連合大会, G12-37, p. G275 (2003.11).

(6) 槌田謙太郎, 岡宏憲, 一階良知, 薦田憲久: “ジョブショップ問題における計画修正ルート付きロバストスケジュールの作成”, 電気学会情報システム研究会, IS-04-32, pp. 7–10 (2004.6).

(7) 岡宏憲, 槌田謙太郎, 一階良知: “タブーサーチによる環境変化に対応する修正解評価期待値を用いたスケジューリング方式”, 電気学会情報システム研究会, IS-05-32, pp. 25–30 (2005.8).

(8) 中條宏昭, 岡宏憲, 一階良知, 薦田憲久: “リアルタイム生産スケジューリングにおけるアトラクター選択適用の検討”, 電気学会情報システム研究会, IS-05-35, pp. 43–48 (2005.8).

(9) 中條宏昭, 岡宏憲, 一階良知, 薦田憲久: “アトラクター選択を用いたリアルタイム生産スケジューリング方式”, 平成17年電気関係学会関西支部連合大会, G11-4, p. G232 (2005.11).

(10) 中條宏昭, 岡宏憲, 一階良知, 薦田憲久: “アトラクター選択を用いたリアルタイム生産スケジューリング方式”, 電気学会システム・制御研究会, SC-05-26, pp. 39–44 (2005.12).

内容梗概

近年，企業は製品供給の低コスト化や顧客満足度の向上のために，生産者から消費者まで製品を供給するためのネットワークであるサプライチェーンを効率化することが求められ，それに応えるためにサプライチェーンマネージメントについての議論がなされている。

サプライチェーン効率化の手段の1つとして，サプライチェーンを構成する各企業において受発注交渉や生産計画作成などの意思決定を自動化することにより，人手の作業を軽減したり，意思決定を高速化・高品質化することが挙げられる。しかし，それぞれの企業が個別に最適化を進めたのでは，部品の納入を急がせるなど他の企業に無理を強いることになり，サプライチェーン全体を見た場合に効率的でない場合がある。両者が許容できる範囲内で受発注を行い，それに合わせた生産計画を作成すべきである。そして，ある製品の受注はその部品の発注を発生させ，サプライチェーン全体に伝わってゆく。また，どこかで輸送遅延などが発生すれば，その影響はそれより下流の企業へと波及する。したがって，サプライチェーンにおいて意思決定を行う上で，サプライチェーン全体・直接取引のある企業・企業内という粒度の異なる3つのレベルを縦断的に取り扱う必要がある。

以上の観点から，本研究では (1) サプライチェーン全体を分析・評価するためのモデリング・シミュレーションツール，(2) 企業間の受発注交渉を計算機同士で自動的に行う方式，(3) 生産計画作成における大規模ジョブショップ問題に対する統合タブーサーチシステム，を提案する。

本論文は全5章から構成される。

第1章の序論では，サプライチェーンマネージメントの重要性を述べ，関連研究および本研究の位置付けを示す。

第2章では，サプライチェーン全体のモデリング・シミュレーションツールに関する研究について述べる。構成要素数が膨大で構造が複雑であるサプライチェーン全体をいかに効率よくモデル化するかという問題に対し，サプライ

チェーンを機能毎に分割してモデル化する手法を提案し，それを用いたモデリング・シミュレーションツールを開発する。そして提案手法を実データおよび検証のための実験用データに適用し，その有効性を評価する。

第3章では，企業間の受発注交渉の自動化に関する研究について述べる。受発注において，各企業は自らの要求を満たす生産計画を作成できるように交渉を行うが，計画問題には生産コストなど企業が社外に出すことを好まない情報が含まれる。この交渉を計算機同士で自動的に行うために，まず，問題をモデル化する。そして，知識を用いて互いに相手の案を修正しあうことにより，各自の計画問題の詳細を相手に秘匿した状態で両者の要求を満たす合意案を自動的に作成する方式を提案する。提案手法を検証のための実験用データに適用し，その有効性を評価する。

第4章では，大規模ジョブショップ問題に対する統合タブーサーチシステムに関する研究について述べる。一般的なジョブショップ問題は，ジョブと呼ばれる製品製造に必要な複数の処理工程の集合を，処理を実行する複数の機械資源に割り当てる問題であり，さらに本研究ではジョブ数が約2000と大規模である点，同一処理工程を必要とするジョブをまとめて処理しなければならない点，専門家が経験的に評価している要素が多数存在するため評価関数の完全な定式化が困難である点などの特徴的な課題がある。そこで，近傍解作成時に適用するルールを確率的に変更するタブーサーチエンジンを核に，探索の並列化，パラメータ自動調整，専門家による解修正の機能を統合したシステムを提案する。そして提案手法を実データに適用し，その有効性を評価する。

第5章では，結論として本研究で得られた成果を要約し，今後の課題について述べる。

目 次

第1章 序論 **1**

1.1 本研究の背景 1

1.2 関連する研究 4

1.2.1 SC全体のモデリング・シミュレーションツール 4

1.2.2 受発注交渉の自動化手法 5

1.2.3 グループ制約付JSPを対象としたスケジューリング手法 6

1.3 本研究の方針 7

1.3.1 SC全体のモデリング・シミュレーションツール 7

1.3.2 二者間交換分配交渉方式 8

1.3.3 パラメータ自動調整機能付き対話型スケジューリングシステム 8

1.4 本論文の構成 9

第2章 サプライチェーンシミュレーションのための機能別モデル定義方式とツール **11**

2.1 緒言 11

2.2 機能別モデル定義方式 12

2.2.1 アプローチ 12

2.2.2 機能別モデル定義の課題 13

2.2.3 機能別モデル定義方式の概要 14

2.2.4 部品モデル 15

2.2.5 物理モデル 16

2.2.6 情報モデル 18

2.2.7 意思決定モデル 19

2.2.8 需要モデル 21

2.2.9 ノードとアークが備える演算機能 21

2.3 モデリング・シミュレーションツール 25
2.3.1 ツールの概要 25
2.3.2 モデル定義機能 26
2.3.3 計画関数登録機能 27
2.3.4 シミュレーション機能 28
2.4 評価実験 29
2.4.1 モデル定義機能の評価 30
2.4.2 シミュレーション精度の検証 32
2.4.3 モデル変更の容易性の評価 34
2.5 結言 35

第3章 二者間交換分配交渉における目的分析知識を用いた交渉方式 37
3.1 緒言 37
3.2 既存手法の問題点 38
3.3 モデル化と交渉方式のアプローチ 39
3.3.1 アプローチの概要 39
3.3.2 モデル化 40
3.3.3 交渉方式 42
3.4 目的分析知識を用いた交渉方式 42
3.4.1 交渉方式の概要と交渉に用いる知識 42
3.4.2 合意案探索のアルゴリズム 44
3.4.3 初期部分解の導出 45
3.4.4 暫定部分解の導出 45
3.4.5 交渉案の改良 47
3.5 評価実験 49
3.5.1 問題の構造 49
3.5.2 使用する知識 51
3.5.3 問題設定1の交渉経過と結果 52
3.5.4 問題設定2の交渉経過と結果 54
3.5.5 考察 56
3.6 結言 57

第4章　グループ制約付き大規模ジョブショップ問題向きの確率的ルール適用方式による統合タブーサーチシステム　59

4.1　緒言 59

4.2　グループ制約付き JSP 60

4.2.1　問題概要 60

4.2.2　計画の評価 62

4.3　確率的ルール適用方式による統合タブーサーチシステム 63

4.3.1　提案システムの構成 63

4.3.2　基本スケジューリングエンジン 64

4.3.3　並列スケジューリングモジュール 67

4.3.4　パラメータ自動調整モジュール 69

4.3.5　対話型解修正モジュール 73

4.4　評価実験 77

4.4.1　実験対象とパラメータ 77

4.4.2　並列タブーサーチモジュールの評価 81

4.4.3　パラメータ自動調整モジュールの評価 82

4.4.4　解修正モジュールの評価 84

4.5　結言 87

第5章　結論　89

5.1　本研究のまとめ 89

5.2　今後の研究課題 90

謝辞　93

参考文献　95

第1章

序論

1.1 本研究の背景

原料調達，製品製造，物流，販売といった，生産者から消費者まで製品を供給するために必要なネットワークをサプライチェーン（Supply Chain，以下SC）という。SCは複数の企業で構成され，個々の企業には部品を調達するための調達部門，生産を担当する生産部門，物流を担当する物流部門，受注などを担当する販売部門などが存在する[1, 2, 3]。このようにSCには複数の企業，部門が関わり合っており，利害が複雑に絡むため，部門間，企業間での連携を行うのが困難な状況である。そのため，これまでの企業努力においても，部門毎または企業毎に個別に効率化を図るにとどまっていた[4, 5, 6]。しかし，近年企業はますます多様化する製品を低コストで提供することが求められている。そのニーズに応えるために，最新の情報通信技術を意思決定や情報の管理・伝達・共有に用いることにより，部門毎，企業毎の効率化のみならず，複数企業にまたがった効率化が試みられるようになった。そのような中で，要求されたサービスレベルを満たしながらSC全体のコストを最小化することを目的として，供給者や生産者，卸売店や小売店を，適切な量の製品が，適切な場所に，適切な時間に生産・配送されるよう効果的に結合するためのサプライチェーンマネージメント（Supply Chain Management，以下SCM）が注目を集めている[7, 8, 9, 10, 11, 12, 13]。

SCMを考える際には，その適用範囲として大きく分けて次の3つの粒度が考えられる。

(1) SC 全体

(2) 直接取引のある企業

(3) 企業内

(1) の SC 全体について，SC 全体を対象とした SCM ソフトはまだ存在せず，企業による実践はまだ行われていないが，その重要性は認識されてきている。しかし，理論的に SC の効率化を行うことは，以下のような理由により困難である。まず，SC を構成する企業や扱われる製品などは SC 毎に異なっているため，ある SC での効率化の方法を，他の SC にそのまま適用することはできない場合が多い。また，SC は多くの不確定要素を含んでいる。部品到着の遅延や機械の故障，道路状況による輸送の遅延などは日常的に起こりうることであり，SC の最適化においては，これらの不確定要素を考慮しなければならない。さらに，SC で扱われる製品は 1 種類だけではなく，その多くがある製品とそれに伴う派生製品からなるプロダクトファミリと呼ばれる製品群を扱っている場合が多い。同じプロダクトファミリに属する製品は，部品や生産に要する処理が共通であるものと独自のものが混在するため，その製品群の物流や生産工程は複雑なものとなる。このような SC の特性から，SC を包括的に効率評価するための手法としてシミュレーション [14, 15] による解析が有効であると言える。

(2) の直接取引のある企業について，企業間で行われる受発注業務においては，受注から納品までに関わる組織間で，これまでの紙による各種伝票を電子化することによって，取引のペーパレス化，データの再入力を省力化し，その正確化をはかり，ビジネスのスピード化を目指した EDI（Electronic Data Interchange，電子データ交換）が普及してきている [16]。また，ERP パッケージ（Enterprise Resource Planning Package，統合基幹業務パッケージ）や EDI などの業務パッケージソフトウェア企業は，これら各組織のシステム間でデータ連携可能となるよう提携して開発を進めている [2, 17, 18]。しかし，いくらデータの入力や交換が効率化されても，受発注者双方が大きな利益を得られる交渉案の作成は人間が行わなければならない。そのような交渉案の作成も含めた，受発注業務の自動化が望まれている。受発注において行われる交渉は交換分配交渉 [19] に分類される。交換分配交渉とは，物やお金といったリソースの分配や交換を行う際に，評価関数の最大化や制約条件充足など個別の要求

を持った交渉者間で，各交渉者の要求を満たすようなリソースの移動案を作成する問題であり，ゲーム理論における非対称情報・協調交渉ゲーム [20, 21] に位置付けられる。商品の価格など1変数の値を二者間で決定する1対1バーゲニングモデルや，複数の買い手の中から1人の買い手と価格を決定するオークションモデルなど，シンプルな問題における戦略と解の関係はゲーム理論を用いてほぼ完全に解明されている [22, 23, 24, 25]。しかし，通常の受発注業務においては価格だけでなく商品数や納期など依存関係を持つ複数の変数の値を決定する必要があり，その複雑性のためゲーム理論による均衡解の解析は困難である。

(3) の企業内について，企業内の効率化においては，各部門毎に別々に構築されていたシステムを統合して相互に参照・利用できるようにするためのERPパッケージと呼ばれるソフトウェアの導入が進められている。これにより，財務会計や人事などデータの一元管理，システムのバージョンアップや保守点検の容易化，他部門の作業のリアルタイムな参照などが可能になる [2, 17]。しかし，各企業において扱う製品やその生産工程，意思決定の方法などはさまざまに異なり，部門毎に独自のノウハウを持っているため，細部にまで汎用のソフトウェアを適用することはできない。特に生産の日程計画作成においては，製品や生産設備により固有の制約や評価基準が存在するために一般的なスケジューリング手法を用いることができないことがあり，その場合には，専門家が知識と経験に基づき手作業で行っている。手作業による計画作成には多大な時間がかかるため，計算機を用いて効率化することが望まれている。特に，いくつかの必要処理工程からなる複数のジョブを，それらの処理を行うためのラインと呼ぶさまざまな機械資源に割り当てる問題が生産計画によく現れる。このような問題をジョブショップ問題またはジョブショップ・スケジューリング問題（Job-shop Scheduling Problem, 以下 JSP）と呼ぶ。JSP は組み合せ最適化問題であり，産業における応用分野が広いために古くから盛んに研究されており，制約条件や評価関数がシンプルなベンチマーク問題が考案され利用されている [26, 27, 28, 29, 30, 31]。しかし，現実には問題に固有の複雑な制約などが存在する。そのような制約の一つにグループ制約がある。グループ制約とは，同一処理工程が必要な複数のジョブはあるサイズの範囲内でグループ化し，グループ毎に処理を割り当てなければならないという制約である。グループ構成はさまざまな組み合わせが可能であるため，従来研究されてきたシンプ

ルな JSP よりも組み合せ数が膨大になる。

以上述べた SC の 3 つの粒度における本研究の位置付けを図 1.1 に示す。本論文では，図 1.1 に示すように，SC の各粒度において効率化を図るために，SC 全体の効率を評価するためのモデリング・シミュレーションツール，企業間で行う受発注交渉の自動化手法，および，企業内での生産計画作成の効率化のためのグループ制約付 JSP を対象としたスケジューリング手法の実現を課題として取り上げる。

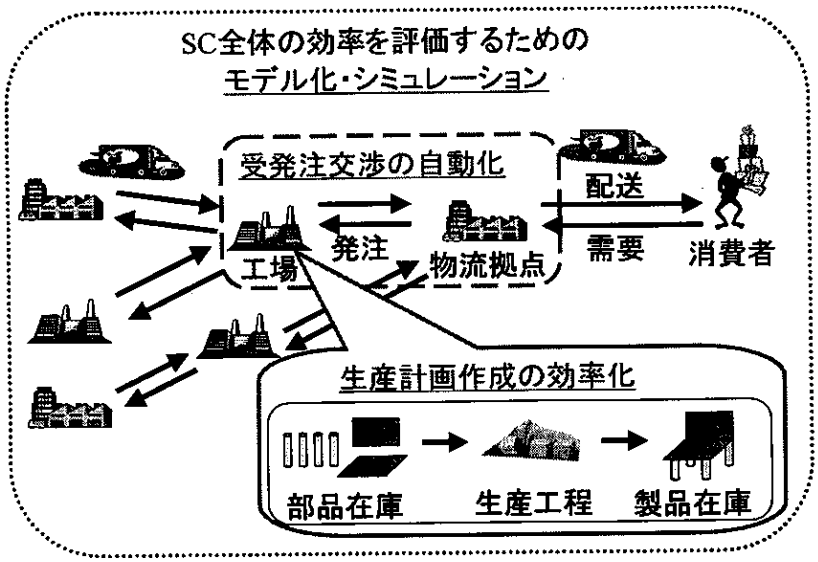

図 1.1: 本研究の位置付け

1.2 関連する研究

以上述べた課題に対して，本節では，SC 全体を評価するためのモデリング・シミュレーションツール，企業間で行われる受発注の交渉を計算機同士で自動的に行うシステム，および，グループ制約と呼ばれる制約を持った JSP を対象としたスケジューリングシステムに関する従来の研究について述べる。

1.2.1 SC 全体のモデリング・シミュレーションツール

SC に関する従来の学術的な研究では，生産や配送，在庫管理といった SC モデルの一部分のみを取り上げ，その他の部分は切り捨てるか極端に抽象化した

理論的な研究 [32, 33, 34, 35] がなされており，SC 全体を対象としたものは見当たらない。

SCM コンサルティングを行っている企業において SC を評価する場合には，SC のシミュレーションも行われている。その場合，モデル化する対象を絞り込み，C 言語などの汎用言語によってその都度プログラミングしたり，ExSpect[36] などの汎用のシミュレーションツールを使用している [37]。このため，最初に SC をモデル化するのに時間がかかる上，モデルを変更する際にもソースコードやモデル定義画面から該当箇所を探すなど，煩雑な作業が必要である。

Oracle 社の "Oracle Supply Chain Planning" など商用の SCM ソフトには SC のシミュレーション機能を持っているものも存在する。しかし，それは生産計画・配送計画・調達計画といった部分的スケジュールを立案するためのもの [38] であり，SC 全体を視野に入れたシミュレーションを行うことはできない。また，SC モデルは SCM ソフト導入時にソフト提供業者が顧客企業向けに構築するのが一般的であり，その後に顧客企業が大幅な変更を加えることを想定したような機能は SCM ソフトには見当たらない。

したがって，SC 全体の効率を評価するためには汎用の言語やツールを用いてシミュレーションを行う以外に手段が無いのが現状である。

1.2.2 受発注交渉の自動化手法

第 1.1 節で述べたように，受発注において行われる交渉は交換分配交渉に分類される。複数の変数の値を決定する交換分配交渉に適用可能な解法の研究として，各交渉者の制約条件をひとまとめにする解法 [39] やリソース移動案の評価値を交渉者間でやり取りする解法 [40] がある。例えば，Rubenstein-Montano らの解法 [40] では，交渉者それぞれが，各リソースの重要度を表す重みと入手したい理想的な数量を設定し，リソース移動案と理想値との差と重みの積和を評価関数とする。そして，全交渉者が評価値を示しあい，ある交渉者の評価値を改善するためには他の交渉者の評価値を改悪しなければならないような Pareto 最適 [41] なリソース移動案を選択する。この解法では，重みの大きなリソースの数量だけが評価基準となり，全交渉者が同じ構造の評価関数を持つ。

しかし，実際の受発注における交渉では，交渉者はリソースの移動案が制約の一部となるような計画問題を持っている。例えば，製品を製造する企業とそ

の部品を製造する企業との間で行われる部品売買の交渉を考えると，製品製造企業は製品の生産計画を持っており，交渉の結果入手できる部品の量や納期が，生産計画を作成する際の制約条件となる。また，そのリソースの移動案に基づいて作成した計画案に対して，工場の稼働率を最大化したいなどの評価関数が存在する。部品製造企業にも同様に，販売量や納期が制約となる在庫引当・補充の計画問題があり，コスト軽減などの評価関数が存在する。この計画問題を解かなければリソース移動案を評価できず，また評価関数は交渉者毎に異なるため，Rubenstein-Montano らの手法は適用できない。一方，計画問題の部分的な制約であるリソースの移動案から，計画問題の評価関数を算出するには時間がかかるので，提案されたリソース移動案の候補を評価するために度々計画問題を解くことを繰り返すアプローチは現実的ではない。

また，各交渉者の計画問題をひとまとめにすれば，一つの計画問題として数理計画法 [42, 43, 44] などの既存手法で解くことが可能である。しかし現実の交渉問題，特に企業間の交渉では，それぞれの計画問題の詳細な内容（種々の制約条件や要求の内容）を交渉相手に示すことは好ましくない場合が多く，それらをひとまとめにはできない。

1.2.3 グループ制約付 JSP を対象としたスケジューリング手法

組み合せ最適化問題の最適解を求める手法として，分枝限定法 [45, 46, 47] や動的計画法 [48, 49, 50, 51] などがある。しかし，JSP は単に NP（Nondeterministic Polynomial time，非決定性多項式時間）困難であるばかりでなく最適解を求めることが非常に難しい問題とされており，実用的な時間で最適解を得ることは非常に困難である。そこで，最適性は保証されないが高速に解を得ることができるメタヒューリスティクスとして，自然界において淘汰により環境に適した遺伝子が残るプロセスを模倣した遺伝的アルゴリズム（Genetic Algorithm, 以下 GA）[52, 53, 54, 55, 56, 57] や，解のサイクリング（探索がいくつかの解を経由して既に発見した解に戻ること）を防ぎながら近傍集合中の最良解の探索を繰り返すタブーサーチ [58, 59, 60, 61, 62] を JSP へ適用することが試みられている [63, 64, 65, 66, 67, 68, 69, 70, 71] が，その対象はベンチマーク問題などシンプルな JSP である。

グループ制約を持つ JSP に GA を適用する場合，既存研究と同様に遺伝子

の各要素を1つの処理として解をコーディングしたのでは，グループ制約に違反した致死解が多数発生するという問題がある。一方，遺伝子の各要素を1つのグループとして解をコーディングする場合，適切なグループ構成を選択する手法が必要となる。GAの制約として遺伝子のサイズは全ての個体で等しくなければならないため，グループ数の異なる解を同時に探索することはできず，また，次世代の個体を作成する際の遺伝的操作は単純な操作であり，本問題に特化した効果的な操作は行いにくい。

それに対し，タブーサーチでは解の表現方法や現在の解を改善して近傍集合を作成するためのルールを柔軟に定義可能であるため，グループ制約を持つJSPにも比較的対応しやすいと言える。グループ制約を持つJSPにタブーサーチを適用する研究として，大前らは，ジョブ自由度という概念を用いてヒューリスティクスにより高速に初期解を生成し，得られた初期解に対して改善のために適用するルールを確率的に変更するタブーサーチを適用することで，効率的にグループ計画を立案する手法[72, 73]を提案している。しかし，探索速度および得られる解の質が不十分であるとの専門家の意見がある。

1.3 本研究の方針

前節までに述べた課題をふまえて，SCのそれぞれの粒度における効率化を支援する各システムの研究方針を示す。

1.3.1 SC全体のモデリング・シミュレーションツール

本研究で提案するSCモデリング・シミュレーションツールの使用目的は，SC全体のモデルの修正とシミュレーションを繰り返し行うことで，試行錯誤的に優れたSC構成を調査することである。そのため，新規SCモデルの作成だけでなく既に作成されているモデルの部分的変更も効率的に可能，すなわち少数の構成要素・短時間の作業によってモデルの作成・変更が可能なツールを開発することが本研究の目的となる。

そこで，SCを機能に基づき部分モデルに分割して定義する方式を提案し，効率的なモデル定義を支援する種々の機能を備えたモデリング・シミュレーションツールとして実装する[74, 75, 76, 77]。

1.3.2 二者間交換分配交渉方式

製品が消費者に届くまでには，小売店，卸問屋，製品製造企業，部品製造企業など，複数の企業を通り，そのそれぞれの間で受発注が行われる。上流の企業に何をどれだけ発注するかは，下流の企業から何をどれだけ受注したかに影響される。しかし，通常は確定している受発注の内容や予測される需要に合わせて次の新たな受発注契約のための交渉を開始するのであり，一度確定した受発注契約の内容を他企業との交渉の状況によって変更することは一般に行われない。また，複数の顧客から同時期に発注を受けるような場合も，一件ずつ個別に対応するのが一般的である。そこで，本研究では二者間での交換分配交渉のみを対象とする。

そして，非対称情報（交渉者毎に制約条件などが異なる）かつ不完全情報（互いに相手の制約条件などを参照不可能）という条件下で，協調して両者の制約条件を同時に充足する解（リソース移動案）をいかに発見するかということに焦点を当て，二者間交換分配交渉における問題のモデル化と解法の提案を行う [78, 79, 80, 81]。

交渉者間でやり取りする情報は必要最小限の内容とする。現在普及しているEDI の標準データフォーマットは業界別に標準化されており統一されていないため，提案手法では独自のフォーマットによりリソース移動案を表現する。リソース移動案に記述する各項目はあらかじめ交渉者間で共通の認識を持っているものとし，その方法については本研究では扱わない。

1.3.3 パラメータ自動調整機能付き対話型スケジューリングシステム

本研究の対象は，ラインに割り当てなければならないジョブの数が 2000 程度，ジョブを割当可能なラインの数が 10 程度という大規模なグループ制約付き JSP であり，専門家からは「評価関数を完全に定式化することは困難であるため最終的な解は目で確認した上で修正したい，その修正も含めて 2 時間以内に解を得たい」という要求がある。これに対し，大前らの手法 [72, 73] では，専門家が 8 時間かけて手作業で作成した解に対して評価値で比較して 95%の解を 2 時間程度で得られている。この手法をベースに，タブーサーチを並列化

し，また適切なタブーサーチのパラメータを短時間で得るための自動調整方式を開発することにより高速化し，専門家による解の修正のための時間を確保する。また，専門家が解を修正するために，(1) 得られた解のガントチャート表示，(2) その解に対する要求の入力，(3) その要求を満たす解の探索，を繰り返し行う対話型システムを提案する [82, 83]。

1.4 本論文の構成

本論文では，第 2 章以降を以下のように構成する。

第 2 章では，文献 [74, 75, 76, 77, 84] に基づき，SC モデル定義方式とそれを用いたシミュレーションツールについて述べる。機能に基づき分割した各部分モデルをそれぞれの機能に即した記述方法によって効率的に定義する方式を提案する。複数の部分モデルに重複して現れる構成要素や頻繁に変更する可能性が高い計画関数群はツールによって管理し，ユーザの負担を軽減する。実データおよび検証のための実験用データに基づくモデル化とシミュレーションの実験を行い，提案ツールの有効性を確認する。

第 3 章では，文献 [78, 79, 80, 81] に基づき，二者間交換分配交渉問題の解法について述べる。リソース移動案（交渉案）に対する評価関数は用いず，制約充足問題としてモデル化する。各交渉者は交渉案の作成方法の指針を記した目的分析知識を使用することにより，自らの条件だけでなく相手の条件も充足する可能性の高い交渉案を作成し，互いに提案しあう。提案手法を検証用の交渉問題に適用し，合意案を発見可能であることを確認する。

第 4 章では，文献 [82, 83] に基づき，グループ制約を持つ JSP を対象とした統合スケジューリングシステムを提案する。まず従来手法 [72, 73] を説明し，それを拡張する 3 つの手法を提案する。従来手法を並列化し，各探索にはそれぞれ異なる乱数列と初期解を用いることで，切り分けられた探索空間を同時に探索する。その際のルール適用確率は，効果的なルールを多用するよう事前に自動調整する。また，専門家が解を修正する際には入力に応じてルール適用確率を動的に変更し，要求を満たす解を作成する。実験により，これらの手法を統合することによって目標が達成されることを確認する。

第 5 章では，結論として本研究で得られた成果を要約し，今後の課題を述べる。

第2章

サプライチェーンシミュレーションのための機能別モデル定義方式とツール

2.1 緒言

企業は多様な製品を低コストで供給することが求められている。そのためには，各企業・各部門が業務の効率化に努めるだけでなく，サプライチェーン（SC）全体を効率化することも重要である。本章では，SCM コンサルティングにおいて使用することを想定した，構成が異なる SC 全体の効率を比較して評価するためのツールを提案する。

SC の構成要素は数が膨大であり，構造が複雑である。また，SC 上で行われる活動は意思決定から情報伝達，製品の生産・輸送まで多岐に渡り，需要変動や事故など確率的に発生する事象も存在する。このように複雑な SC の評価・改善を行うには，シミュレーションが適している。しかし，従来は汎用のシミュレーションツールやプログラミング言語を使用しなければならず，SC モデルの作成・変更に長時間を要するという問題がある。

本章では，SC 全体のモデル作成・変更を効率的に行うためのモデル定義方式を提案し，モデリング・シミュレーションツールを開発する [74, 75, 76, 77, 84, 15]。SC には，生産計画作成のような論理的機能や，実際の生産のような物理的機能がある。そして，現実の SC において低コストで実現可能な，設備増強など

の物理的な変更を伴わない論理的機能の改善策がシミュレーションモデルの変更においても試みられることが多い。そこでSCを複数の機能別部分モデルに分割し，変更される頻度が大きい部分モデルを上位層として階層化し他の部分モデルに変更の影響が及ばないようにすることにより，効率的なモデル変更を可能とする。各部分モデルは属性を持ったノードとアークによってフローの構造を直観的に表現する。時間の経過や在庫数の増減などの演算機能はあらかじめノードとアークに具備しておき，ユーザはそのパラメータを属性として設定するだけでシミュレーション可能とする。複数の部分モデルに重複して現れる構成要素は部分モデル間の不整合性や冗長な定義によるモデル定義時間の増加の原因となるため，提案手法を実装したツールによって管理する。また，SCのシミュレーションによる評価値として，在庫・生産・輸送の各コストと納期達成率を出力する。

以下，第2.2節では機能別モデル定義方式について，アプローチと各部分モデルの詳細を述べる。第2.3節ではそれを実装したモデリング・シミュレーションツールの機能ついて述べる。第2.4節では評価実験を行い，提案ツールによるモデル作成・変更の効率とシミュレーション結果の精度を確認する。

2.2 機能別モデル定義方式

2.2.1 アプローチ

シミュレーションによって改善効果が確認されたSCの変更方法が現実のSCに反映されることを考えると，現実世界において実現が容易な変更が，シミュレーションモデルの変更においても試みられやすいと考えられる。実現が容易なSCの変更とは，必要となる費用が小さく，また，変更箇所が少ないような変更である。生産設備などSCの物理的機能の変更には大きな費用が必要であり，また，それに伴い意思決定アルゴリズムなど論理的機能の変更も必要になる。それに対し，SCの論理的機能は，物理的機能に影響を与えることなく小さな費用で変更することが可能である。そこで，SCのシミュレーションモデルを複数の機能別部分モデルに分割し，変更頻度の大きい部分モデルを上位層として階層化することで，モデル変更時の変更箇所を少なくする。また，部分モデル毎に，それぞれの機能に即したモデル記述方式を用意し，各モデルを簡

潔に定義可能とする。

SCの物理的機能においては，製品間の構成関係（Bill of Material, 以下BOM）の変更が複数の生産設備に影響を与えることから，BOMのモデルを最下位層，企業施設のモデルを上位層として分割する。SCの論理的機能においては，企業間での情報連携は当事者間での調整が必要であり，意思決定アルゴリズムのような企業内のみの変更と比較して実現しにくい。また，企業が持つ情報が変更されると，その情報を利用するための意思決定アルゴリズムの変更が必要になる。したがって，SCの論理的機能において，企業情報のモデルを下位層，意思決定のモデルを上位層として分割する。これらに，原則として供給側から操作できない需要を合わせ，SCのシミュレーションモデルを，次に示す5つの機能別の部分モデルに分割する。

- 部品モデル: BOMを表す。静的なデータのみで表現するモデルである。
- 物理モデル: SCの物理面の構成要素について，企業施設や顧客の位置などの静的なデータ，および，製品等の物理的な物の流れなど動的なデータを持つモデルである。シミュレーション時には，輸送遅延などの不確定要素や意思決定モデルから伝達された命令に応じて，物理的な活動を行う。
- 情報モデル: SCの情報面の構成要素について，どの企業がどの施設を所有するかといった静的なデータ，および，受発注等の情報の流れを表す動的なデータを持つモデルである。シミュレーション時には，需要モデルで発生した需要や意思決定モデルから伝達された命令に応じて，発注などの情報の流れを表現する。
- 意思決定モデル: 企業の意思決定アルゴリズムを表すモデルであり，シミュレーション時には，物理モデル，情報モデルに命令を伝達する。
- 需要モデル: 顧客の需要発生を表すモデルである。

2.2.2 機能別モデル定義の課題

SC全体を機能別に分割してモデル化する上で，次の2点が課題となる。

- 複数の部分モデルに共通するSCの構成要素の存在

  このような構成要素を各部分モデルにおいて個別に定義する場合，部分モデル間の不整合性，冗長な定義によるモデル定義時間の増加を避ける必要がある。

- 各部分モデルの，機能に即した簡潔な記述方法

  製品や情報のフロー表現，需要発生の確率表現，計画関数の手続き表現など，各部分モデルの機能毎に表現しなければならない内容は異なる。その一方で，SCモデル全体を容易に把握・記述するためには全ての部分モデルにおいて記述方法が簡潔な方法で統一されていることが望ましい。

2.2.3 機能別モデル定義方式の概要

部分モデル間の整合性を持たせるために，複数の部分モデルに重複する構成要素は1つの部分モデルで一元管理し，他の部分モデルでは管理元のモデルを参照することとする。これにより，構成要素を一元管理している部分モデルのみの変更によって，その構成要素を参照している全ての部分モデルが同様に変更される。このような参照操作はツールによって実現する。

また，各部分モデルは，製品の部品構成やフローの構造を直観的に表すために，モデル毎に特有の属性を持つノードとアークで表現する。属性の例としては，物流拠点を表すノードの属性として製品在庫数，拠点間の輸送経路を表すアークの属性として輸送所要時間，顧客を表す顧客ノードの属性として需要発生間隔の確率分布関数のパラメータなどが挙げられる。

そして，時間の経過や確率的に発生する事象の制御，SC活動による属性値の変化，部分モデル間のデータ伝達など，シミュレーションに必要な機能はノードとアークにあらかじめ持たせておくことにより，それらの機能をツールのユーザが記述する手間を省く。また，改善されたSCモデルを作成することがツールの使用目的であるため，SCの評価値として納期達成率や各種コストを計算する機能が必要である。

以上をまとめ，各部分モデルが持つ機能および部分モデル間の関係を図2.1に示す。図中の破線は部分モデル間の静的データの参照関係を，実線はシミュレーション中の動的データの流れを，それぞれ示している。部品モデルは最下

位層であるため他の全ての部分モデルから参照され，また，静的なデータしか扱わないため演算機能は持たない。物理モデルは部品モデルを除き他の全ての部分モデルから参照され，また，生産や輸送のための演算機能を持つ。情報モデルは意思決定モデルと需要モデルから参照され，また，受発注のための演算機能を持つ。意思決定モデルは最上位層の部分モデルの１つであり，物理モデルや情報モデルの動作をコントロールするための演算機能を持つ。需要モデルもまた最上位層の部分モデルの１つであり，顧客の需要発生に関する演算機能を持つ。第2.2.4節から第2.2.9節で，各部分モデルの記述方法，および，各ノードとアークが持つ機能について説明する。

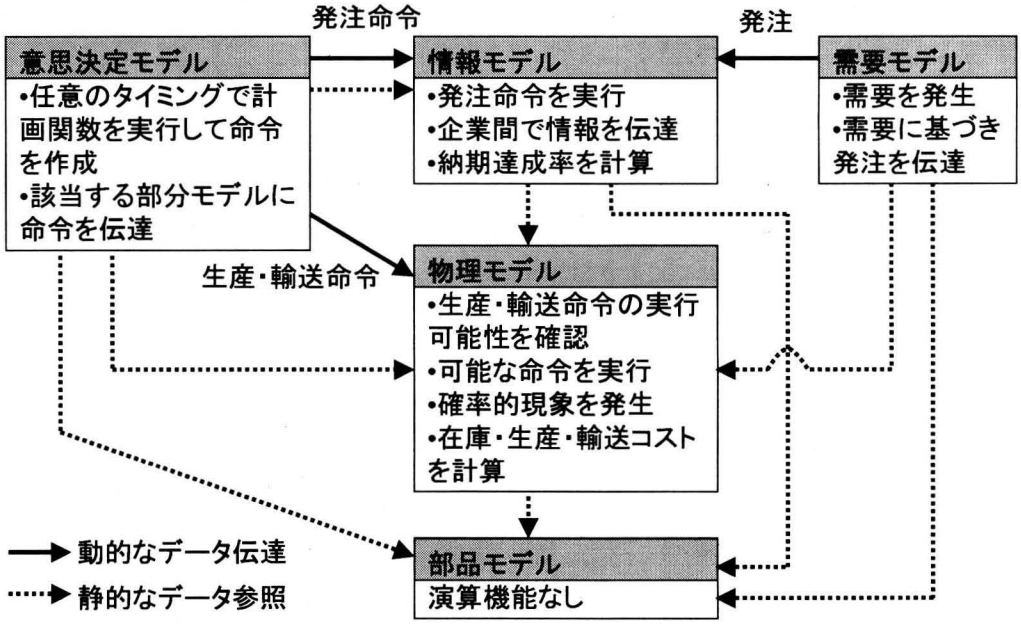

図 2.1: 各部分モデルの機能と参照関係

2.2.4 部品モデル

部品モデルは，図2.2に示すように，部品や組立品に相当するアイテムと，アイテム間の関係を表す工程を構成要素とする。各構成要素が持つ属性を以下に示す。

- アイテム（長方形のノード）：アイテム名
- 工程アーク（実線のアーク）：工程名，部品の必要数

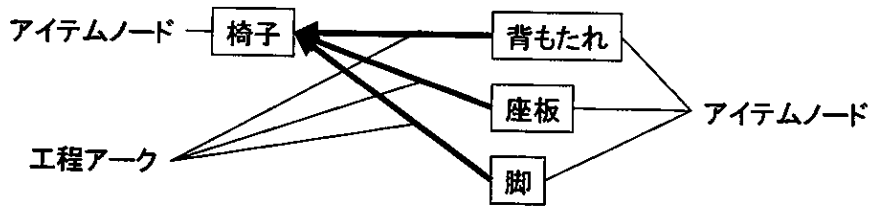

図 2.2: 部品モデルのグラフ表現

ただし，実際に図 2.2 のようなグラフ表記を用いる場合には，1 つの工程を表現するために同じ工程名を持つ複数の工程アークを記述する必要があり，特にモデルを変更する際に間違いが発生する可能性が大きい。また，1 つのアイテムノードに複数の工程アークが接続され，視認性が悪くなることが考えられる。そこで，モデリングツールにおいてユーザは部品モデルのグラフ表記を直接扱わず，各行が 1 つの工程アークに相当する表を工程毎に作成する。この表の属性は，工程アークおよびその工程アークと接続されるアイテムノードの属性である。工程名と組立品のアイテム名は，ツールにより 1 行目のデータが全ての行に自動的に反映されるものとする。図 2.2 の椅子組立を表で表現したものを表 2.1 に示す。

表 2.1: 部品モデルにおける工程の表表現

工程名	組立品アイテム名	部品アイテム名	部品アイテム必要数
椅子組立	椅子	背もたれ	1
椅子組立	椅子	座板	1
椅子組立	椅子	脚	4

2.2.5 物理モデル

物理モデルは，企業施設や顧客の位置，およびそれらの間での物の流れを表現する。また，シミュレーション実行時には，SC モデルの評価指標として，在庫・輸送・生産のためのコストを計算する。これらのコストは，複数の SC モデルの相対的な評価を行うための値であり，厳密な金額を計算する必要はない。そのため，コスト計算のためのパラメータは，在庫や輸送・生産工程を定義するためのノードやアークの属性として，次に示す 3 種類の値を設定することと

する。

- 在庫コスト: 在庫品目毎に，アイテムを1個，1日在庫するのに要するコストを指定
- 輸送コスト: 輸送手段毎に，アイテムを1個，1回輸送するのに要するコストを指定
- 生産コスト: 工程毎に，アイテムを1ロット生産するのに要するコストを指定

SCにおける物理的な物の流れは，工場や物流拠点，消費者の間での輸送と，各工場における生産工程に分けられる。そのため，物理モデルは図2.3に示すように，輸送モデルと生産工程モデルの2階層に分けて定義する。

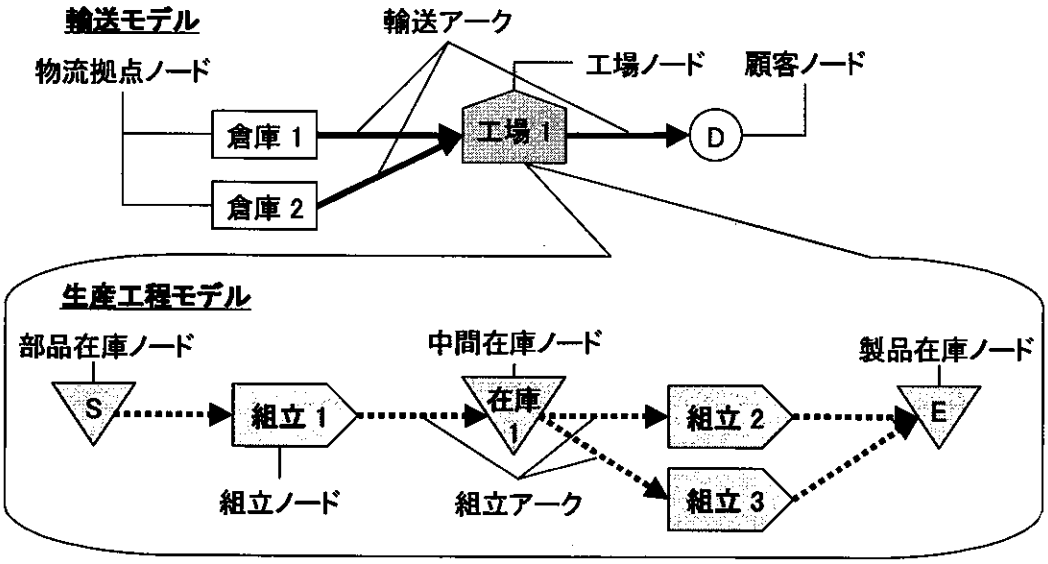

図 2.3: 物理モデル

2.2.5.1 輸送モデル

輸送モデルは，工場や物流拠点，消費者の間での輸送経路や，物流拠点に存在する在庫量などを表現する。輸送モデルの各構成要素が持つ属性を以下に示す。

- 工場ノード（家型のような五角形のノード）：工場名，生産工程モデル

- 物流拠点ノード（長方形のノード）：物流拠点名，在庫品目（部品モデルのアイテムを参照して選択，複数選択可能），初期在庫量，在庫コスト
- 顧客ノード（円形のノード）：顧客の物理的な位置
- 輸送アーク（実線のアーク）：輸送手段名（複数選択可能），輸送時間，輸送コスト，遅延確率，遅延時間（遅延確率および遅延時間は正規分布などの確率を用いて表現）

2.2.5.2 生産工程モデル

生産工程モデルは，各工場内での生産工程を表す。在庫ノードは，部品在庫用と製品在庫用の他，必要に応じて中間生産物用の中間在庫ノードが用いられる（図 2.3 参照）。生産工程モデルの各構成要素が持つ属性を以下に示す。

- 在庫ノード（逆三角形のノード）：在庫名，在庫品目（部品モデルのアイテムを参照して選択，複数選択可能），初期在庫量，在庫コスト
- 組立ノード（右矢印のような五角形のノード）：組立ノード名，生産能力，故障確率，故障時間，生産可能工程（部品モデルの工程を参照して選択，複数選択可能，ロットサイズ，生産コスト，必要生産能力も定義）
- 組立アーク（点線のアーク）：属性なし

2.2.6 情報モデル

情報モデルは企業ノード，顧客ノード，発注アークによって構成し，これらの接続関係により，図 2.4 に示す企業間および企業と顧客の間での情報の流れを定義する。情報モデルの顧客ノードは，物理モデルの顧客ノードと 1 対 1 に対応する。また，流れる情報の種類や各企業がどの施設を所有しているかという情報を，以下のような各構成要素の属性として定義する。

- 企業ノード（かまぼこ断面型のノード）：企業名，所有施設（物理モデルの物流拠点，工場を参照して選択，複数選択可能，ただし重複所有不可）

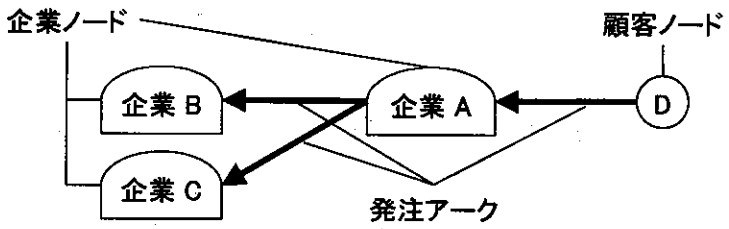

図 2.4: 情報モデル

- 発注アーク（実線のアーク）：発注可能品目（部品モデルのアイテムを参照して選択，複数登録可能），提供情報（文字，数値など）
- 顧客ノード（円形のノード）：属性なし

2.2.7 意思決定モデル

意思決定モデルは企業の意思決定アルゴリズムを表し，シミュレーション時には他の部分モデルへ生産や輸送，発注の命令を出す。情報モデルの企業ノードを参照し，1企業につき1つの意思決定モデルを定義する。図2.5に示すように，意思決定アルゴリズムの計画実行単位として意思決定ノードを用意し，意思決定ノード毎に生産計画や輸送計画などの計画関数を対応付ける。計画関数とは，受注や在庫の状況などを入力とし，それを基に作成した生産や輸送の計画を出力する関数であり，手続き的に記述する。ただし，本定義方式では計画関数の記述方法は限定せず，第2.3節で示すように，ツールでは計画関数自体は任意のプログラム言語で記述した外部プログラムとして実装する。シミュレーション時には，意思決定アークによりトランザクションが伝達されたときに計画関数を実行し，トランザクションはタイマノードにより定期的に，またはイベントノードにより生産設備の故障や在庫の状況といった特定の状況時に発生させる。

意思決定ノードで起動された計画関数の出力を，生産命令や輸送命令，発注命令として物理モデルや情報モデルに伝達するために，生産命令ノード，輸送命令ノード，発注命令ノードを用意し，データベースアーク（以下，DBアーク）により意思決定ノードと接続する。DBアークはデータを伝達するためのアークである。生産命令ノードに伝えられた生産計画（生産開始時刻，工程名，

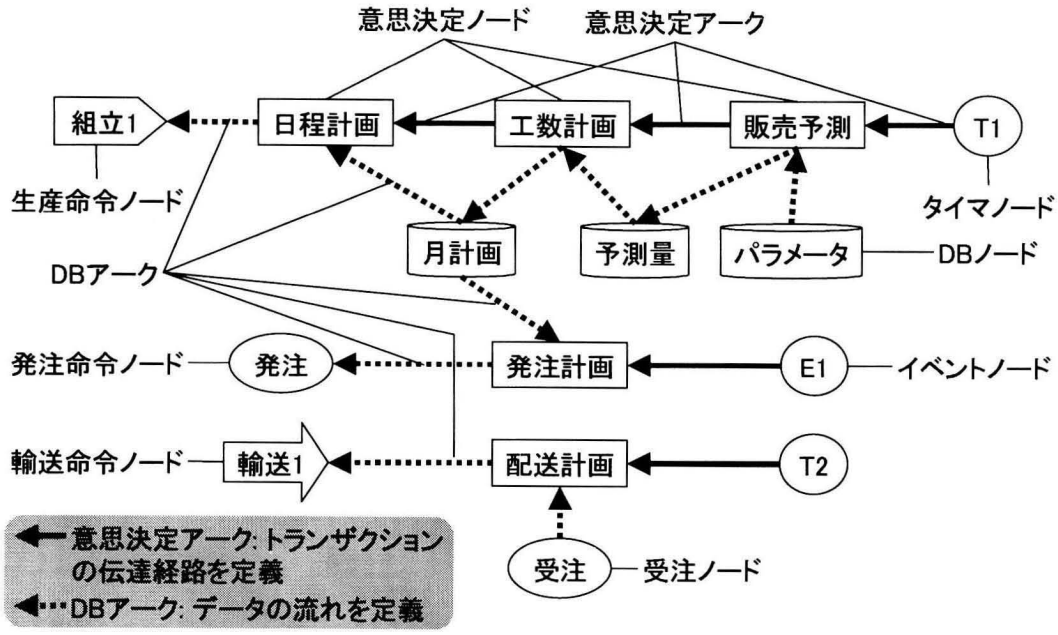

図 2.5: 意思決定モデル

生産ロット数），輸送計画（輸送開始時刻，輸送品目，輸送量，輸送手段名）は，それぞれ生産命令，輸送命令として物理モデルへ，また，発注命令ノードに伝えられた発注計画（発注先企業名，発注品目，発注量，納期，納入先施設名）は発注命令として情報モデルへ，それぞれ伝達し，実行させる。生産命令ノード，輸送命令ノードは，それぞれ物理モデルの組立ノード，輸送アークと1対1で対応する。また，他企業からの受注のデータ（発注元企業名，発注品目，発注量，納期，納入先施設名）が格納されるノードとして受注ノードを用意し，受注ノードと意思決定ノードをDBアークで接続することにより，受注データを計画関数の入力とする。発注命令ノードと受注ノードは，それぞれ情報モデルの企業ノードと1対1で対応する。その他任意のデータを計画関数の入出力として使用するために，データベースノード（以下，DBノード）を用意する。

各構成要素が持つ属性を以下に示す。

- 意思決定ノード（長方形のノード）：意思決定名，計画関数（計画関数ライブラリから選択，第2.3.3節参照）
- 意思決定アーク（実線のアーク）：起動回数，間隔（アーク元の起動1回

に対してアーク先が起動される回数，間隔）

- DB ノード（低い円筒形のノード）：DB 名，DB の持つ属性（複数定義可能，属性名，属性値の型も定義）
- DB アーク（点線のアーク）：属性なし
- 生産命令ノード（右矢印のような五角形のノード）：属性なし
- 輸送命令ノード（右矢印型のノード）：属性なし
- 発注命令ノード（楕円の中に発注と書かれたノード）：属性なし
- 受注ノード（楕円の中に受注と書かれたノード）：属性なし
- タイマノード（円の中に T と書かれたノード）：起動開始時間，間隔
- イベントノード（円の中に E と書かれたノード）：監視する生産工程名または在庫名，物流拠点名（監視品目，イベントを発生させる基準在庫量も定義，複数定義可能）

2.2.8 需要モデル

需要モデルでは，情報モデルで定義された顧客ノードに対して，発注品目，需要量，需要発生間隔，納期といった属性を設定し，需要の発生パターンを表現する。需要量や需要発生間隔，納期は，正規分布，指数分布などの確率を用いて表現する。

2.2.9 ノードとアークが備える演算機能

需要モデルにおける需要の発生および意思決定モデルにおける命令の作成，情報モデル・物理モデルでの命令の実行という処理の流れに沿って，シミュレーションを行うためにノードとアークが備える演算機能を，表 2.2 から表 2.5 に示す。

表 2.2: モデルのノードとアークが備える演算機能（その 1）

機能名	機能の説明
需要発生	需要ノードで定義された需要パターン毎に，定義された需要発生間隔にしたがって需要を発生させる時刻を確率計算する。需要発生時刻には，需要量，納期を確率計算する。
タイマノードにおけるトランザクション発生	タイマノードにおいて，定義された発生間隔にしたがってトランザクションを発生させる。
イベント監視（故障）	組立ノードで故障が起こった場合に，そのノードがイベントノードでの監視対象のノードである場合，該当するイベントノードでトランザクションを発生させる。
イベント監視（在庫）	物流拠点ノード，在庫ノードにおいて，在庫の減少が起こった場合に，そのノードがイベントノードでの監視対象のノードであり，かつイベントノードで定義された基準在庫量を下回った場合，該当するイベントノードでトランザクションを発生させる。
意思決定アークにおけるトランザクション伝達	意思決定アークで定義された間隔，回数にしたがい，タイマノード，イベントノードにおいて発生したトランザクションを，アーク先の意思決定ノードに伝達する。
意思決定ノードの起動	トランザクションが意思決定アークから意思決定ノードに到達した場合に，計画関数を起動する。
受注ノードの参照	意思決定ノードで起動された計画関数に，受注ノードの受注データを引数として渡す。
DB ノードの参照	意思決定ノードで起動された計画関数に，意思決定ノードが参照する DB ノードのデータを引数として渡す。
DB ノードへの書込	意思決定ノードで起動された計画関数の戻り値を，意思決定ノードから接続された DB ノードへ書き込む。
発注命令ノードへの書込	意思決定ノードで起動された計画関数の戻り値である発注計画（発注品目，納期，数量，発注先などのデータ）を発注命令ノードへ書き込む。

表 2.3: モデルのノードとアークが備える演算機能（その 2）

機能名	機能の説明
生産命令ノードへの書込	意思決定ノードで起動された計画関数の戻り値である生産計画（組立品目，数量などのデータ）を生産命令ノードへ書き込む。
輸送命令ノードへの書込	意思決定ノードで起動された計画関数の戻り値である輸送計画（輸送品目，輸送先などのデータ）を輸送命令ノードへ書き込む。
発注命令の伝達	発注命令ノードに書き込まれた発注計画を，その発注命令ノードに対応する企業ノードへ発注命令として伝える。
生産命令の伝達	生産命令ノードに書き込まれた生産計画を，その生産命令ノードに対応する組立ノードへ生産命令として伝える。
輸送命令の伝達	輸送命令ノードに書き込まれた輸送計画を，その輸送命令ノードに対応する輸送アークへ輸送命令として伝える。
発注	需要ノードで発生した需要および企業ノードが受け取った発注命令を実行し，発注先企業の受注データに追加する。
発注命令ノードの発注計画削除	発注した製品が相手企業から輸送された場合に，発注命令ノード中の該当する発注計画を削除する。
受注ノードの受注データ削除	受注した製品が相手企業に輸送された場合に，受注ノード中の該当する受注データを削除する。
納期達成率計算	発注命令ノードの発注計画が削除された場合に，削除された発注計画中の納期と，発注計画が削除された時刻を比較して納期が守られたかを調べ，企業の納期達成率を再計算する。
組立ノードの故障確率計算	組立ノードで定義された故障確率にしたがい，故障発生時刻を計算する。故障が起これば，故障の間は生産能力を 0 にする。
組立ノードの故障時間計算	組立ノードで定義された故障時間にしたがい，故障が発生した組立ノードにおいて故障終了時刻を計算する。故障終了時刻を過ぎたら生産能力を元の値に戻す。

表 2.4: モデルのノードとアークが備える演算機能（その 3）

機能名	機能の説明
生産におけるロット計算	生産命令を実行する場合に，生産指示されたロット数を製品個数に換算する。
生産における生産能力チェック	生産命令が実行される場合に，組立ノードの生産能力データにアクセスし，生産に必要な生産能力が残っていれば生産命令を実行する。生産能力不足ならば実行しない。
生産における在庫チェック	生産命令が組立ノードで実行される場合に，アーク元の在庫のデータにアクセスし，部品となる在庫が生産に十分な量であれば生産命令を実行する。在庫不足ならば実行しない。
生産における生産能力使用	生産命令を実行したときに，組立ノードの生産能力データにアクセスし，生産に必要な生産能力を組立ノードの生産能力から減算する。
生産における部品使用	生産命令を実行したときに，組立ノードに接続されたアーク元の在庫データにアクセスし必要な部品を減算する。
生産における製品出力	生産命令を実行したときに，組立ノードに接続されたアーク先の在庫データにアクセスし，生産した製品を加算する。
生産コストの計算	生産命令を実行したときに，組立ノードで組み立てられた工程の生産指示分のロットサイズと生産コストを掛け合わせ企業の生産コストに加算する。
輸送における製品搬出	輸送命令を実行したときに，輸送アーク元の在庫データにアクセスし，輸送した製品を減算する。
輸送における遅延確率計算	輸送命令を実行したときに，輸送アークで定義された遅延確率を計算し，遅延すると決定した場合には遅延時間の演算を呼び出す。
輸送における遅延時間計算	輸送遅延を発生させた場合に，輸送アークで定義された遅延時間を計算し，輸送の時間経過演算に対して加算する。
輸送における時間経過	輸送命令を実行したときに，輸送アークで定義された輸送時間だけ時間を経過させたのち，輸送における製品出力の演算を呼び出す。

表 2.5: モデルのノードとアークが備える演算機能（その 4）

機能名	機能の説明
輸送における製品出力	輸送時間経過演算の後に，輸送アーク先の在庫データにアクセスし，輸送した製品を加算する。
輸送コストの計算	輸送命令が行われた場合に，輸送アークで定義された輸送手段の輸送コスト分を企業の輸送コストに加算する。
在庫コストの計算	物流拠点ノード，部品在庫ノード，製品在庫ノード，中間在庫ノードにおいて，在庫に存在する品目毎に在庫量と在庫コストを掛け合わせ，単位時間毎に企業の在庫コストに加算する。

2.3 モデリング・シミュレーションツール

2.3.1 ツールの概要

第 2.2 節で述べた機能別モデル定義方式に基づき，SC をモデル化，シミュレーションするツールを開発した。機能別モデル定義方式で記述方法を指定していない意思決定モデルの計画関数は，本ツールでは外部プログラムとして実装する。このため，ユーザが使い慣れたプログラム言語で計画関数を作成可能であり，新たな言語を習得する必要が無い。また，作成した計画関数は計画関数ライブラリに登録して管理し，意思決定モデルにおいて容易に使用可能とする。

図 2.6 に示すように，本ツールはモデル定義機能と計画関数登録機能，シミュレーション機能を持ち，メイン画面と，メイン画面から呼び出す各機能専用の画面によって構成する。モデル定義機能は部分モデル毎の定義画面からなる。計画関数登録機能は計画関数ライブラリ画面からなる。シミュレーション機能は，シミュレーションを制御するためのシミュレーションメイン画面と，シミュレーション中の物理モデル・情報モデル・意思決定モデルそれぞれの動作を確認するための各画面からなる。これらの画面は並列に表示・操作可能とする。第 2.3.2 節から第 2.3.4 節において，各機能の詳細を説明する。

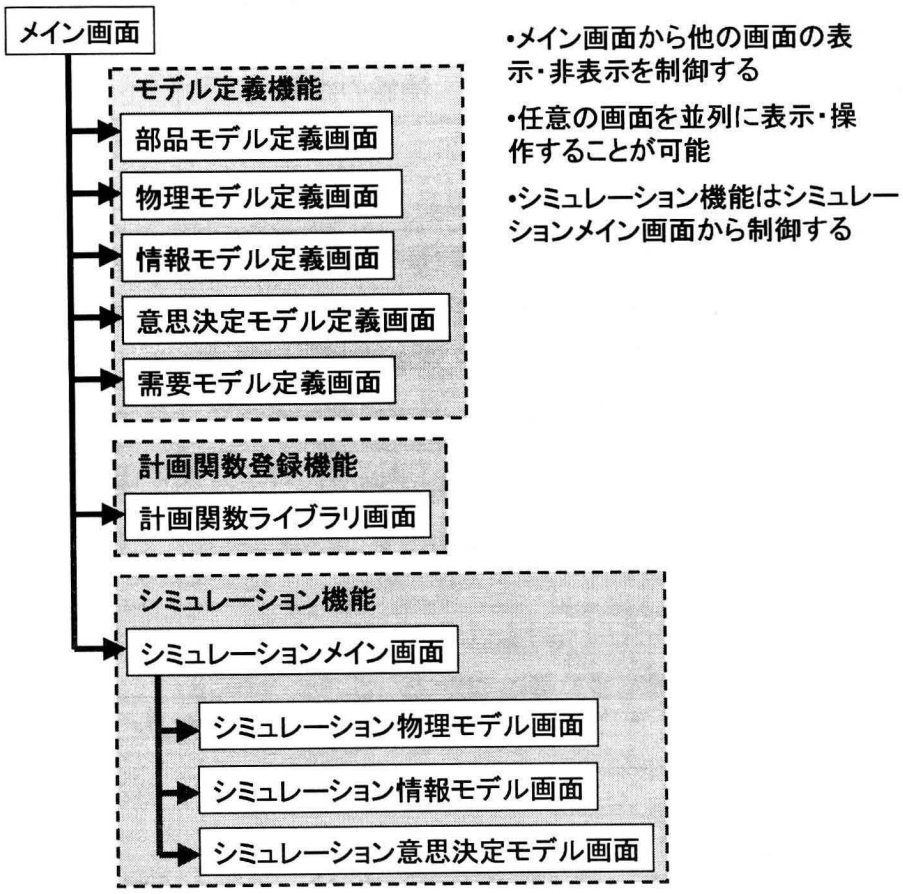

図 2.6: ツールの画面の遷移図

2.3.2 モデル定義機能

部分モデル毎に異なった定義画面を提供し，ある部分モデルの定義の途中においても別の部分モデルの定義画面を随時呼び出し可能とする。具体的には，図 2.7 に示すように，機能別モデルを統括するメイン画面（親画面）上に，部分モデル定義画面（子画面）を複数同時に表示可能とする。これにより，ユーザの意図に合わせ，複数の部分モデルを並列に作成，変更することが可能である。

複数の部分モデルに重複して現れる構成要素を管理元の部分モデルから参照して定義する場合には，管理元のモデルで定義している構成要素の一覧を表示し，その中から選択することで定義を行う。図 2.7 に，情報モデルにおける企業ノードの属性として所有施設を定義する際の事例を示す。ここでは，物理モ

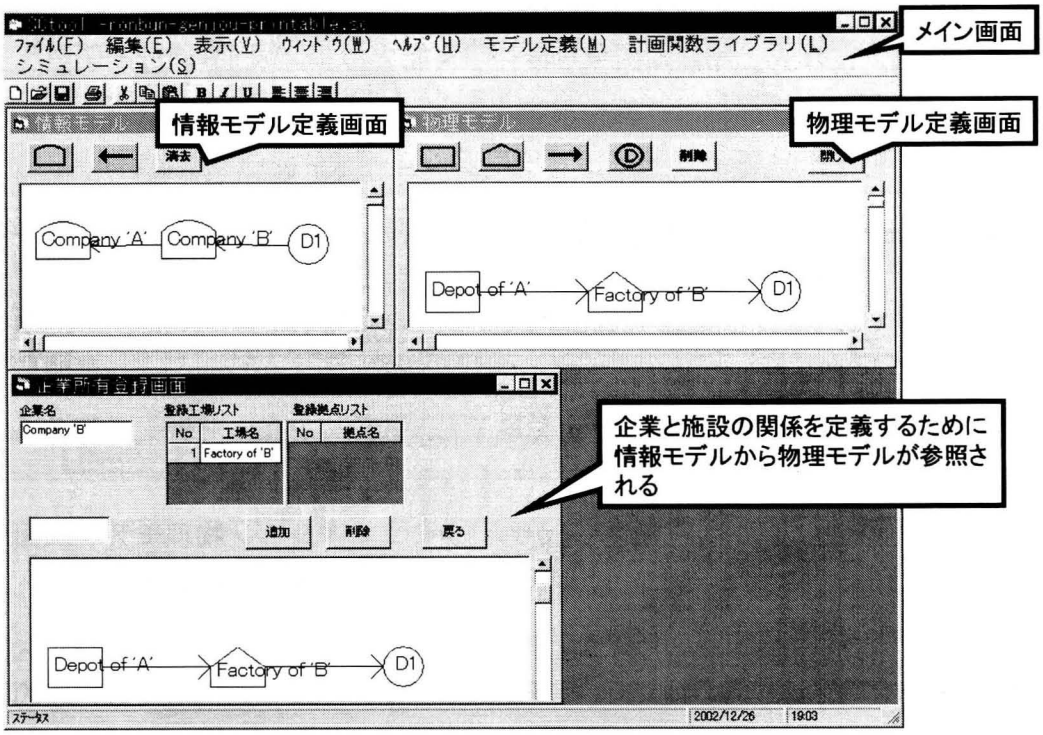

図 2.7: メイン画面と部分モデル定義画面

デルで定義した物流拠点ノード，工場ノードを表示・選択させる画面を表示している。また，物理モデル定義画面上に顧客ノードを配置した場合，情報モデルや需要モデルにも，それと対応した顧客ノードを自動的に配置するなどといった自動化により，ユーザが特別に意識しなくとも，整合性のあるモデルが作成可能である。

2.3.3 計画関数登録機能

外部プログラムを計画関数として管理するために，ユーザが計画関数を自由に登録，使用できる計画関数ライブラリを用意する。このライブラリに計画関数を登録する場合には，計画関数登録画面を通して，外部プログラム名，計画関数名，入出力の仕様（意思決定ノードに接続される DB ノードの仕様），計画関数の内容を説明したコメントを一組として入力する。意思決定モデルにおいて意思決定ノードに計画関数を登録する際には，(1) 関数名とコメントを参考にして計画関数ライブラリから計画関数を選択，(2) 指定された入出力の仕

様にしたがいDBノードを接続，(3) DBノードに値を入力，という一連の操作により，過去に作成した計画関数を再利用可能である。図2.8に計画関数ライブラリのメイン画面と登録画面を示す。

2.3.4 シミュレーション機能

図2.9にシミュレーション中の画面例を示す。シミュレーションメイン画面の開始ボタンを押すことにより，モデル化したSCのシミュレーションが開始される。シミュレーションの一時停止や終了も，同様にシミュレーションメイン画面から操作可能である。

また，シミュレーションメイン画面から，シミュレーション物理モデル画面およびシミュレーション情報モデル画面など部分モデル画面を呼び出し，これらの画面上においてシミュレーション中のSCの挙動を随時確認可能である。具体的には，シミュレーション物理モデル画面では，生産や輸送による物理的アイテムの増減を，生産や輸送を行った工場ノードや輸送アークの上に表示する。シミュレーション情報モデル画面では，発注品目や発注量，納期を発注

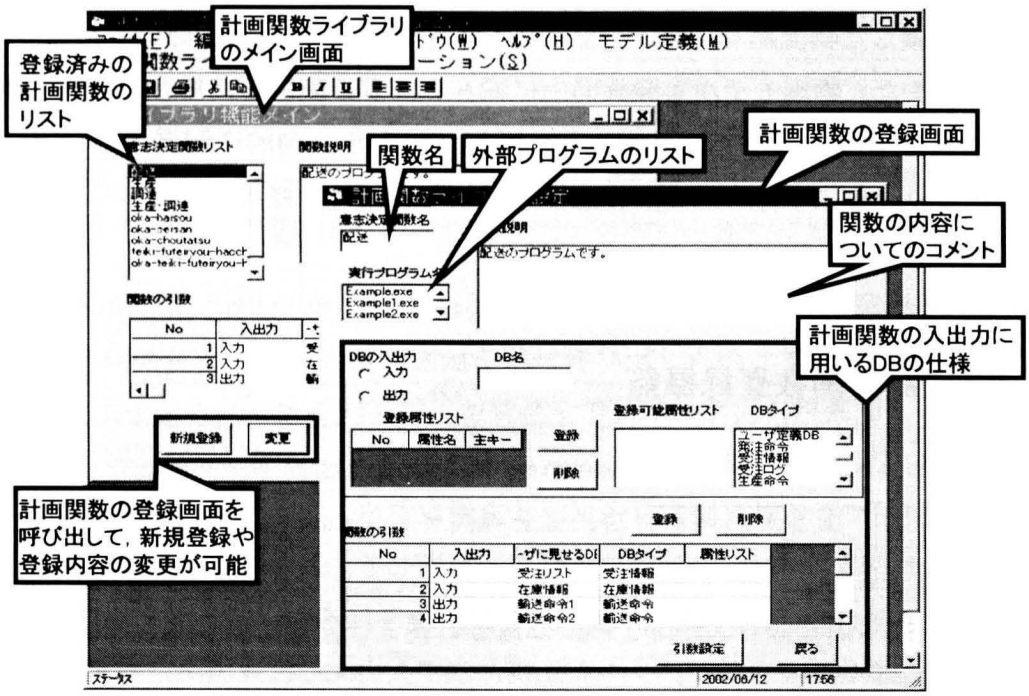

図 2.8: 計画関数ライブラリのメイン画面と登録画面

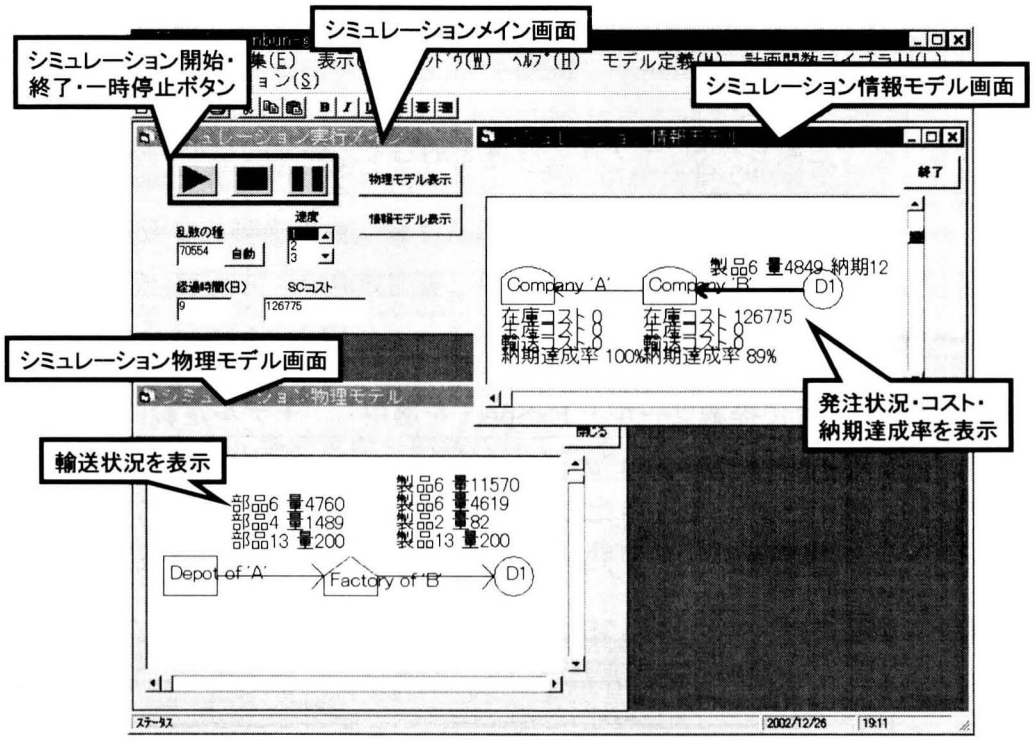

図 2.9: シミュレーション実行画面

アークの上に，また企業毎の在庫コスト，生産コスト，輸送コストの累積，および納期達成率を各企業ノードの上に，それぞれ表示する。シミュレーション意思決定モデル画面では，意思決定トランザクションの発生，伝達の様子や，計画関数の実行によって作成した生産・輸送・発注の命令を，それぞれ該当するノードやアークの上に表示する。さらに，これらの値の時間変化をログファイルとして出力し，表計算ソフトウェアなどを用いた分析を行うことが可能である。

2.4 評価実験

本ツールの有効性を確認するために，3つの実験を行う。

(1) 検証のための SC モデルを，提案ツールと，汎用シミュレーションツールである ExSpect [36, 37] とを用いてそれぞれ作成し，その容易さを比較評価する実験

(2) 実在するSCに対して提案ツールを適用し，シミュレーション結果の精度を評価する実験

(3) 上記(2)で定義したSCモデルの改善を通して，部分的なモデル変更の容易さを評価する実験

2.4.1 モデル定義機能の評価

検証のためのSCに提案ツールとExSpectを適用し，モデル定義に要する時間，構成要素（ノードとアーク）の合計数を比較した。ExSpectはイベント駆動型の汎用モデリング・シミュレーションツールである。提案ツールと同じくノードとアークを用いたグラフィカルな表示方式を持っており，また，独自のプログラム言語によりノード毎に任意の手続きを定義可能である。なお，実験は提案ツールの開発者である筆者が行った。筆者は，比較対象であるExSpectも熟知しており，両ツールの習熟度に差はない。また，この種のツールの利用者として想定されているSCの設計者の習熟度は，筆者と同程度と考えられる。なお，対象SCの設計者も筆者である。

実験対象のSCは，複数の企業間でアイテムの発注・輸送や製造が行われたり確率的現象が発生したりするといった一般性があり，また，提案ツールでモデル化する際に全てのモデリング機能を使用する必要があるようなSCとして，次に示すようなSCを使用した。

このSCは，車輪や肘掛の有無，回転の可否や色が異なる椅子のプロダクトファミリを製造するという組立産業のSCであり，製品・部品を合わせて47種類のアイテムが流れる。このSCには，1つの椅子製造企業，4つの一次部品供給企業，6つの二次部品供給企業が存在し，椅子製造企業と一次部品供給企業はそれぞれ工場を1つ，二次部品供給企業はそれぞれ物流拠点を1ずつ持つ。ただし，二次部品供給企業の物流拠点には出荷用の在庫が常に十分に存在し，受注した翌日に必ず出荷できる。また，各工場，輸送経路において，製造装置の故障や配送遅延が確率的に発生する。また，これらの企業では合計33種類の調達，生産，配送の計画関数を用いる。例えば，椅子製造企業では生産計画として，製品の在庫が基準在庫量を下回らないように生産する，在庫充填型生産計画を用いる。これらの計画関数は，提案ツールではC++言語で実装した。

提案ツールを用いて定義した物理モデルを図2.10に示す。椅子製造企業の工場“chair”が一次部品供給企業の工場“Stand”，“Seat”，“Back”，“Arm rest”から脚や座板，背もたれや肘掛等を取り寄せて椅子を組み立てる。工場“Stand”，“Seat”，“Back”は，それぞれ異なる二次部品供給企業の物流拠点2ヶ所から部品を取り寄せて，脚や座板，背もたれを製造する。

以上述べたSCモデルの作成に要した時間と構成要素数を表2.6に示す。ExSpectを用いた場合と比べ，提案ツールを用いた場合には，モデル作成時間が3.6%，構成要素数が12.5%である。提案ツールを用いることで，部分モデル間のデータ参照機能により冗長な定義が削減されていること，ユーザが演算機能を定義する必要が無く，また，意思決定アルゴリズムが使い慣れたプログラム言語を使用して記述可能であることにより，モデル作成時間が短縮されていることが

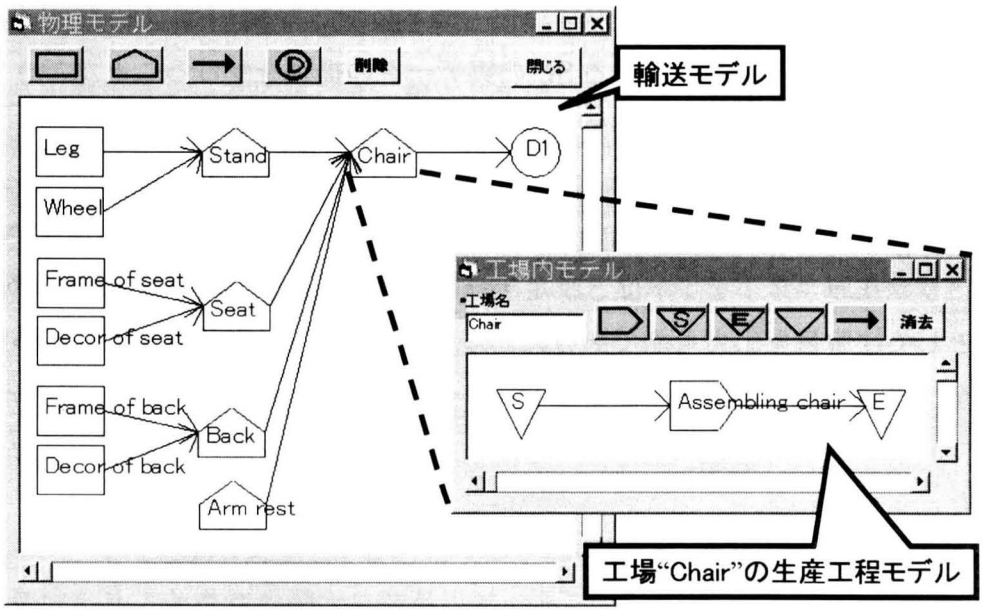

図 2.10: 提案ツールを用いて定義した椅子製造SCの物理モデル

表 2.6: モデル定義実験の結果

使用ツール	モデル作成時間（時間）	構成要素数（個）
ExSpect	42	1364
提案ツール	1.5	171

分かる。したがって，提案ツールのモデル定義機能はSCの容易なモデル化に有効であると言える。

2.4.2 シミュレーション精度の検証

提案ツールはSC改善案の事前評価としての使用を想定しており，シミュレーションにより改善効果が確認されたSC改良案は現実のSCに適用される。その場合には，現実のSCにおいてもシミュレーションと同等の改善効果を求められるため，シミュレーション結果の精度は重要である。そこで，実在するSCを提案ツールによりモデル化・シミュレーションし，その結果を実データと比較して精度を検証した。

この実験のためには，製品のBOMや製造に要するコスト，受発注の実績など，企業が外部に公開することが稀なデータが必要である。しかし，限定的ながら素材産業の物流分野におけるSCの実データの提供を受けることができたので，それを用いて実験を行った。

第2.2.5節で述べたように，モデル化したSCの評価指標として提案ツールが出力する各種コストは相対評価のための独自の単位の値であるが，アイテム1個あたりの在庫コストを1単位と設定すれば，シミュレーション結果の在庫コストとして在庫量を出力可能である。一方，素材産業では在庫量を製品の体積や重量で表現可能であるため，提案ツールの出力と実データを比較しやすい。そこで，在庫量と納期達成率について，シミュレーション結果と実データを比較した。

2.4.2.1 モデル説明

実験対象としたSCは，2つの企業A，B，14種類の顧客からなり，34種類のアイテムが流れる。企業Bは企業Aから17種類の材料（部品）を購入し，17種類の製品（組立品）を生産して顧客に販売する。使用する実データは企業Bのデータであり，BOM，1ヶ月間の製品出庫実績（出庫日，納入先名，出庫品名，出庫合計量），その前月末および当月末，翌月末の在庫量，工場の生産能力（日産最大10000単位），生産計画アルゴリズム（受注後に材料を調達して生産，一部製品は見込み生産），顧客が要求する納期（全て2日），各拠

点間の輸送所要日数（全て1日）である。また，シミュレーションモデル作成のために，データ提供者が妥当だと判断した次の前提をおく。

(1) 生産設備の故障，輸送遅延は発生しない

(2) 企業Bでは，顧客の各受注に対して個別に製品の輸送を行う（顧客の需要は出庫実績と一致する）

(3) 企業Aは常に十分な在庫を所有しており，企業Bの発注に対して即座に製品の輸送を行う

初期在庫量は，企業Aの物流拠点では前提(3)より無限大とし，企業Bの工場では実データにしたがい，製品出庫実績データがある月の前月末の在庫量とする。

需要モデルにおいて発生させる需要は，前提(2)から，企業Bの1ヶ月間の製品出庫実績データに基づき，組立品毎に，以下のように発生間隔と発注1回あたりの数量を設定する。ただし，複数の顧客に納入されている組立品については，顧客毎に独立した需要パターンとして設定する。

- 需要発生間隔: 30/(1ヶ月間の出庫回数) を平均とする，指数分布にしたがう

- 需要量: 平均と分散が，実データの出庫量のそれらと等しい，正規分布にしたがう

全パターンを総合すると，1日あたりの平均需要量は全組立品合計で7409.3単位量であり，企業Bの最大日産量以下である。

企業Bの意思決定モデルでは，「受注に対して十分な量の組立品在庫，部品在庫が存在するならば，生産，輸送の計画を作成し，部品在庫が不足するならば，企業Aに対する不足分の発注の計画を作成する。その後，生産能力に余裕があるならば，見込み生産品を生産する計画を作成する」という計画関数を1日1回起動する。また，企業Aでは，「受注に対して十分な量の在庫が存在するならば，輸送の計画を作成する」という計画関数を1日1回起動する。これらの計画関数の作成には，C++言語を使用した。

2.4.2.2 シミュレーション精度検証実験の結果

300日間のシミュレーションを行い，企業Bの在庫量と納期達成率について，シミュレーション結果が実データと一致する精度を検証した。

(1) 企業Bの在庫量 企業Bの，実際の月末の在庫量データと，シミュレーションによって得られる在庫量を比較する。3ヶ月分の月末の在庫量データによると，全製品の合計の在庫量は，平均 $(5043+4539+4634)/3 = 4738.6$（単位量）である。一方，提案ツールによるシミュレーションの結果，在庫量の平均は4751.6（単位量）であった。使用できる実際の在庫量データが少ないため統計的な検定はできないが，1日あたりの平均在庫量の誤差率は $(4751.6-4738.6)/4738.6 = 2.743\times10^{-3} = 0.274$（%）である。

(2) 企業Bの納期達成率 企業Bの，実際の納期達成率とシミュレーションによって得られる納期達成率を比較する。

モデル化の際の前提1，3より，生産能力を超える量を1日に受注する場合にのみ納期遅れが発生し，前提2より，出庫実績データを受注データとみなすことが可能なので，実際の納期達成率を計算可能である。出庫日毎に各出庫の出庫量を昇順に加算していき，合計量が生産能力以下となる出庫の数を納期達成回数，加算すると生産能力を超える出庫の数を納期遅れ回数とする。こうして実際の納期達成率を計算すると，82.95%である。一方，提案ツールによるシミュレーションの結果は83.33%であり，実際の納期達成率との誤差は0.379ポイントである。

2.4.3 モデル変更の容易性の評価

提案ツールによるモデルの部分的な変更の容易性を評価するために，第2.4.2節の実験で対象としたSCにおいて，需要が増加したために企業Bの生産能力が不足するようになったと仮定し，企業Bの製品販売の機会損失を防ぐためのSC改善案の事前評価を提案ツールを用いて行うという実験を行い，モデル変更に要する時間を測定した。

全ての需要パターンの需要発生間隔が2分の1になったと仮定する。この場合，1日あたりの需要量が倍増して14818.6（単位量/日）となり，企業Bの現

状の生産能力10000（単位量/日）では生産が追いつかず，納期達成率が低下する。そこで，企業Bの工場に，生産能力が5000（単位量/日）の，需要の発生間隔が最小の3製品を生産可能な生産設備を増設することにより，現状の生産設備と合わせて需要量と同程度の生産量となるようにし，需要が増加する以前と同等の納期達成率を目指す。そのためには，次のようなSCモデルの変更が必要である。

(1) 需要モデルにおいて，需要パターンを変更する。

(2) 物理モデルにおいて，企業Bが所有する工場の生産工程モデルに，増設する生産設備を表す生産工程ノードを追加して，組立アークにより部品および組立品を在庫する在庫ノードと接続する。

(3) 並列な2つの生産設備を使用するための計画関数が計画関数ライブラリ内にないため，第2.4.2節の実験に用いた計画関数を改造して新生産設備を優先的に使用するような計画関数を作成し，計画関数ライブラリに登録する。

(4) 意思決定モデルにおいて，意思決定ノードの計画関数を3で作成した計画関数に変更し，また，追加した組立ノードに対応する生産ノードと意思決定ノードを意思決定アークで接続して，増設した生産設備に生産命令が伝えられるようにする。

変更したモデルでシミュレーションを行ったところ，納期達成率は生産設備の増設により，目標通り18.26%から89.75%に改善された。

提案ツール開発者によるモデル変更に要した時間を表2.7に示す。この変更に要した時間は合計で20分程度である。そのうちの15分は計画関数のプログラミングに要した時間であるため，適切な計画関数があらかじめ計画関数ライブラリに用意されている場合には，5分程度で変更可能である。

2.5 結言

本章では，SCシミュレーションモデルを少数の構成要素により短時間で作成，変更可能とする機能別モデル定義方式を提案し，さらに，それを用いたモ

表 2.7: モデルの変更に要した時間

変更した部分モデル	入力した属性数（項目）	入力したノードとアークの合計数（個）	所要時間（分）
需要モデル	27	0	1
物理モデル	12	3	1
意思決定モデル	37	1	2
計画関数の作成	（C++言語で60行の変更・追加）		15
計画関数の登録	（9項目を入力）		1

デリング・シミュレーションツールを開発した。提案ツールにより，汎用ツールと比較して12.5%の時間でモデルを作成可能であること，作成済みのモデルの部分的な変更が短時間で可能であることが確かめられた。また，実在するSCをモデル化してシミュレーションを行った結果を実データと比較したところ，平均在庫量の誤差率0.274%，納期達成率の誤差0.379ポイントという精度が得られた。

一方，提案手法における需要パターンは，確率分布を用いた表現で顧客別・アイテム別に定義するため，現実の受注実績を忠実に再現したシミュレーションを行うのは困難である。実データに基づいた需要モデル作成の支援が望まれる。また，複数の拠点・顧客を巡回しながら集荷・配送を行う配送専門業者や，複数企業での物流システムの共有など，現実のSCをモデル化する上での課題が残っている。

第3章

二者間交換分配交渉における目的分析知識を用いた交渉方式

3.1 緒言

本章では，受発注時に行われる二者間での交渉を計算機同士で自動的に行うための手法を提案する [78, 79, 80, 81]。

サプライチェーン（SC）において企業間で受発注が行われる際には，数量や金額・納入日などを決定するために交渉が行われる。受発注における伝票の電子化やその交換のためのシステムが普及しつつあるが，それらのシステムにおいても交渉を行うのは人間であるため，人的コストが発生する。また，交渉によって部品が安価で調達されれば，それを組み込んだ製品もまた安価で販売可能となり，この連鎖によって，同じSCに属する多くの企業・消費者が製品を安価で調達可能となる。したがって，SCの効率化を考える際にはこの交渉についても考慮するべきである。

受発注時の交渉のように物やお金などのリソースの交換・分配内容を決定するための交渉は交換分配交渉と呼ばれる。交換分配交渉において，各交渉者は交渉結果に依存する固有の計画問題とその計画問題の解に対する要求を持っている。例えば，製品を製造する企業とその部品を製造する企業との間で行われる部品売買の交渉を考えると，製品製造企業は製品の生産計画を持っており，交渉の結果入手できる部品の量や納期が，生産計画を作成する際の制約条件となる。そして，そのリソースの移動案に基づいて作成した計画案に対して，工

場の稼働率を最大化したいなどの要求が存在する。部品製造企業にも同様に，販売量や納期が制約となる在庫引当・補充の計画問題があり，コスト軽減などの要求が存在する。このような両者の要求を同時に満たすリソース移動案を作成することが，交換分配交渉の目的である。このとき，特に企業では内部情報を外部に公開することが好まれないため，お互いに交渉相手の計画問題や要求の内容が不明という条件下で両者の要求を満たすリソース移動案を作成しなければならない。また，交渉者間でやり取りされる情報の内容は少ない方が望ましい。

そこで，本章では，二者間交換分配交渉の問題を評価関数を用いず制約充足問題としてモデル化する。各交渉者のエージェントとなる計算機は同じ交渉システムを持ち，それぞれ自分の合意の条件を充足する交渉案を作成して交渉相手に伝達しあう。交渉案を受け取った交渉者は，それが自分の合意の条件に違反していれば，充足するように改良した交渉案を作成して交渉相手に伝達し返す。この際には，現実の交渉者の経験に基づく知識を用いることにより，交渉相手の合意の条件に違反しない可能性の大きい交渉案を作成する。これを繰り返し行い，一方の交渉者が作成した交渉案がもう一方の交渉者の合意の条件を充足するとき，それを合意案として出力し，交渉を終了する。

以下，第3.2節では交換分配交渉問題に対して既存手法の適用を考えた場合の問題点を述べる。第3.3節ではモデル化とアプローチについて述べる。第3.4節では交渉方式を説明する。第3.5節では評価実験を行い，提案手法によって二者間交換分配問題の解を発見可能であることを確認する。

3.2 既存手法の問題点

Rubenstein-Montanoらは，交渉者それぞれが各リソースについて，そのリソースの重要度を表す重みと入手したい理想的な数量を設定し，リソース移動案と理想値との差と重みの積和を評価関数として，全交渉者が評価値を示しあい，Pareto最適なリソース移動案を選択するという解法を提案している[40]。Pareto最適とは，多目的の計画問題において，ある目的関数の値を改善するためには少なくとも他の1つの目的関数の値を改悪せざるを得ないような状態である[41]。この解法ではリソースの数量しか考慮しないため，各交渉者が持つ個別の計画問題の解に対する要求が満たされるリソース移動案が選ばれるとは

限らない。

リソース移動案の評価値として，数理計画法 [42, 43, 44] などを用いてその移動案に基づき作成した計画問題の最適解（または近似解）の評価関数の値を使用し，Rubenstein-Montano らのようにして Pareto 最適なリソース移動案を選択するという方法も考えられる。しかし，計画問題の種類や規模によっては1つのリソース移動案を評価するだけで膨大な時間がかかるため，この手法は現実的でない。

3.3 モデル化と交渉方式のアプローチ

3.3.1 アプローチの概要

人間が実際に行っている交換分配交渉のプロセスを参考にして，二者間交換分配交渉問題のモデル化と交渉方式の考案を行う。人間の場合，二人の交渉者の間で，一方の交渉者が作成したリソースの移動案の候補やそれに対する「評価」などの限られた情報をやり取りしている。しかし，交渉者独自の評価関数の具体的な数値を交渉相手に示すことは好ましくない。また，評価を現実に関数として定式化することが困難である。そこで，評価関数はおかず，充足すれば高評価となるような各交渉者の満足の目安を表す条件を，交渉の中間目標として交渉者毎に設定する。この条件を制約条件のように扱い，両交渉者の制約条件を全て充足するリソース移動案を探すという，制約充足問題と捉える。評価関数を用いないでモデル化するため，一般的な二人交渉ゲームにおける代表的な妥結点の1つである，両者の利得増加分の積を最大化する Nash 解 [85] や Pareto 最適解などの概念は，本交渉方式には存在しない。本交渉方式の目的は，互いに交渉相手の制約条件が隠匿されているという不完全情報かつ複雑な協調交渉ゲームでの解の発見方式という位置付けである。

満足の目安には2種類あると考えられる。これ以上高評価である必要はないという十分な満足と，これより低い評価の案には妥協できないという妥協の限界である。前者の目安を表す条件を満足条件，後者の目安を表す条件を妥協条件と名付ける。全ての制約条件・妥協条件を満たす計画案が，各交渉者の計画問題の可能解となる。そして，両交渉者が，それぞれの計画問題の可能解のうち，より多くの満足条件を満たす解を導き出すようなリソースの移動案を作成

することを，交換分配交渉の目的とする。

以上の方針より，各交渉者の独自の計画問題を表す制約条件，中間目標を表す妥協条件・満足条件により問題をモデル化する。そして，交渉者間でリソースの移動案の候補およびそれに対する「評価」として「満足」，「妥協可能」，「拒否」をやり取りすることで，問題を解く。

3.3.2 モデル化

二者間交換分配交渉問題を図 3.1 のようにモデル化する。図中の語句の定義は以下の通りである。

- リソース

  リソースとは交渉者間で分配する物，やり取りする物である。分配する

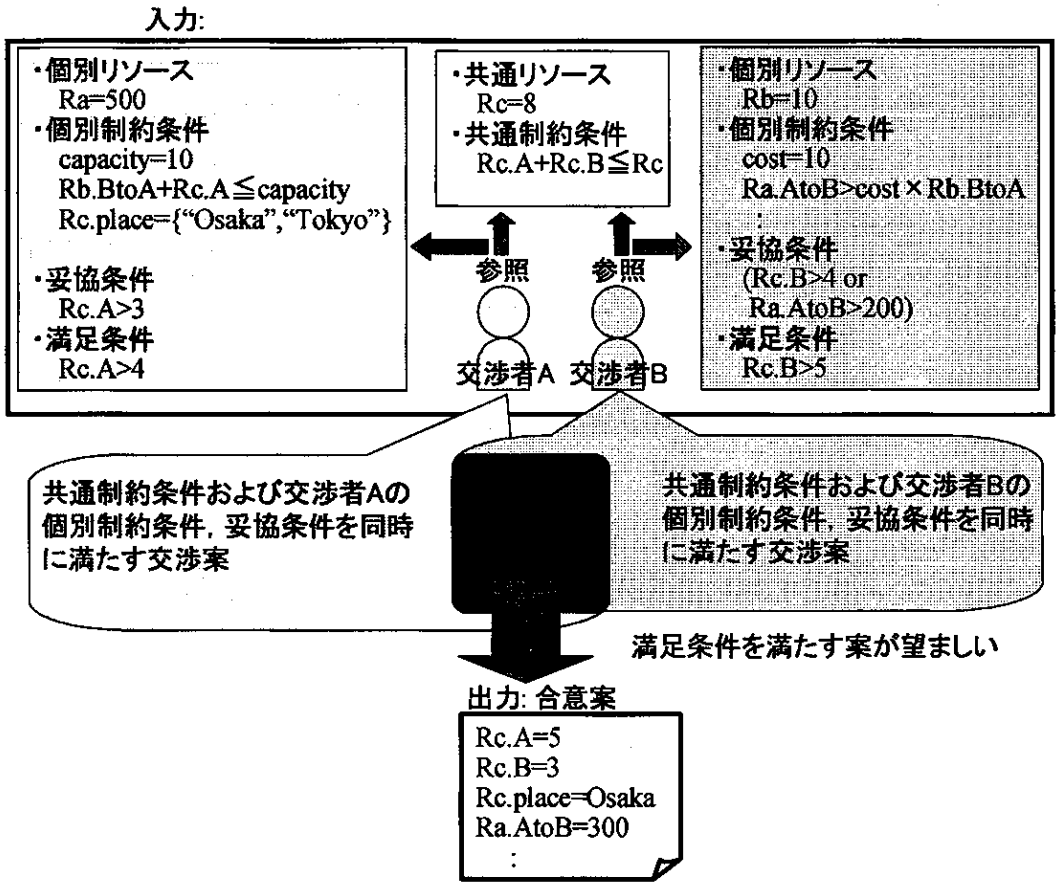

図 3.1: 交換分配交渉問題のモデル化のアプローチ

リソースのことを共通リソース，各交渉者が保持する受け渡し可能なリソースのことを個別リソースと呼ぶ。各リソースはその数量を示す値を持つ。

- 制約条件

  共通リソースの合計量や為替レートのような両交渉者に共通な制約を共通制約条件，各交渉者の計画問題の制約条件で交渉相手には不明なものを個別制約条件と呼ぶ。これら制約条件は，リソースや後に説明する交渉案の属性，各交渉者が持つ個別の変数を引数とする関数の条件式で表現し，論理演算子 “and” または “or” により複数記述することを可能とする。

- 妥協条件

  合意案の評価として最低限の満足の目安を表す条件であり，制約条件と同様に記述する。

- 満足条件

  合意案の評価として十分な満足の目安を表す条件であり，制約条件と同様に記述する。

- 交渉案

  リソースの移動案の候補を交渉案と呼ぶ。次に説明する交渉案の属性の集合で表現する。

- 交渉案の属性

  共通リソースの各交渉者の取り分や個別リソースを受け渡しする日時など，交渉案に記述する，リソースの移動案の内容を表す変数である。

- 合意案

  交換分配交渉問題の解を，合意案と呼ぶ。共通制約条件および両交渉者の個別制約条件・妥協条件を全て充足する交渉案が合意案となり得る。

3.3.3 交渉方式

前述の方針にしたがい，両交渉者が自らの計画問題の可能解を導出するような交渉案を作成し，それを交渉相手に評価させる。交渉案はそれを提案した交渉者の個別制約・妥協条件を充足しているので，交渉相手が作成した交渉案を参考にすれば，交渉相手の制約・妥協条件に違反する可能性・程度が小さい交渉案を作成でき，合意案が作成される可能性が大きくなると考えられる。そこで，交渉案の作成は両交渉者が交互に，それぞれ交渉相手が作成した交渉案を改良して行う。このとき，「たくさん発注すれば単価を安くしてもらえるかもしれない」といった交渉案の属性間の関係に関する知識を利用することにより，効果的な交渉案を作成可能とする。また，交渉案の改良を繰り返しても両者が満足する合意案が得られない場合には，満足条件の違反を許すことで，合意案導出の可能性を高める。

以上のアプローチでは，次の点が課題となる。

- 交渉案の相互改良における，最初の交渉案の属性値の決定
- 交渉案改良時の，値を変更する属性の選択および変更後の値の決定
- 個々の交渉案改良の繰り返しから交渉を収束させるメカニズム

なお，交渉案の条件充足を確認するためには，交渉案からそれに基づく計画問題の解または違反する条件式を出力するような計画機能が必要である。また，交渉案を改良して作成するためには，希望する計画案から逆計画を行い，その計画案が実行可能となるような交渉案の属性値を算出する機能が必要である。しかし，既にさまざまな計画手法が存在するので，本章では各交渉者が自らの計画問題に適した既存手法を用いることを前提として，条件充足確認機能および属性値逆算機能を使用する。

3.4 目的分析知識を用いた交渉方式

3.4.1 交渉方式の概要と交渉に用いる知識

本手法では，一方の交渉者が作成した交渉案を基に，それを相互に改良しあうことを繰り返すことにより合意案を探索する。最初の交渉案を作成する際

には，その交渉者自身の合意の条件を充足するよう属性値逆算機能によって属性値を決定するが，一意に定まらない場合もありうる。そのような場合のために，デフォルトの属性値を知識として用意しておく。また，交渉案を改良する際には，販売価格と販売数のように関係のある複数の属性の値を同時に変更するが，このような属性間の関係は問題に依存し，また，システム上未知である交渉相手の合意の条件に依存する。そのため，ユーザの経験に基づいて，自らの条件だけでなく交渉相手の条件も充足できる可能性が高いような交渉案の改良方法を知識として用意しておく。この知識を目的分析知識と名付ける。目的分析知識は，表 3.1 に示すように，自分が改善を要求する属性（改良属性）1 つとその改善の代わりに相手に妥協する複数の属性（妥協属性），およびそれらの属性値の変更方法（連続値ならば増（+）または減（−），離散値ならば候補となる値の集合）の組として記述する。

表 3.1: 目的分析知識の例

改良属性	妥協属性
Price : −	*Number* : +
Due date : −	*Amount paid* : +
	Item :{Part.P, Part.Q, Part.R}
Amount paid : −	*Item* :{Part.P, Part.Q, Part.R}

目的分析知識を用いて交渉案を改良する際，改良属性は合意の条件の違反を改善するための属性であるので，その値は合意の条件を充足する限界値に変更する。妥協属性は，自らの合意の条件を満たす範囲内で交渉相手に妥協するための属性であるので，自らの合意の条件充足の限界値からの余裕を小さくするよう変更する。ただし，交渉相手に一方的に有利な交渉案とならないよう，妥協条件充足の限界値に向けて徐々に変更してゆく。

また，スムーズに交渉を進めるために，属性をいくつかのグループに分割し，1 グループずつ順に属性を決定する。あるグループの属性を両交渉者の合意の条件に違反しないよう決定すると，そのグループの属性を固定し，次のグループについて交渉を行う。属性のグループ分けは，属性間の関連性の強さやユーザの経験に基づいて行い，グルーピング知識として用意しておく。グルーピング知識は，各属性がどのグループに属するかを，属性名とそれが属するグルー

プ番号の組により全属性に対して記述する。

3.4.2 合意案探索のアルゴリズム

交渉中の交渉案は図3.2のように部分解として扱い，少しずつ属性を追加してゆく。そして，全ての属性が決まったところでその交渉案を合意案とする。

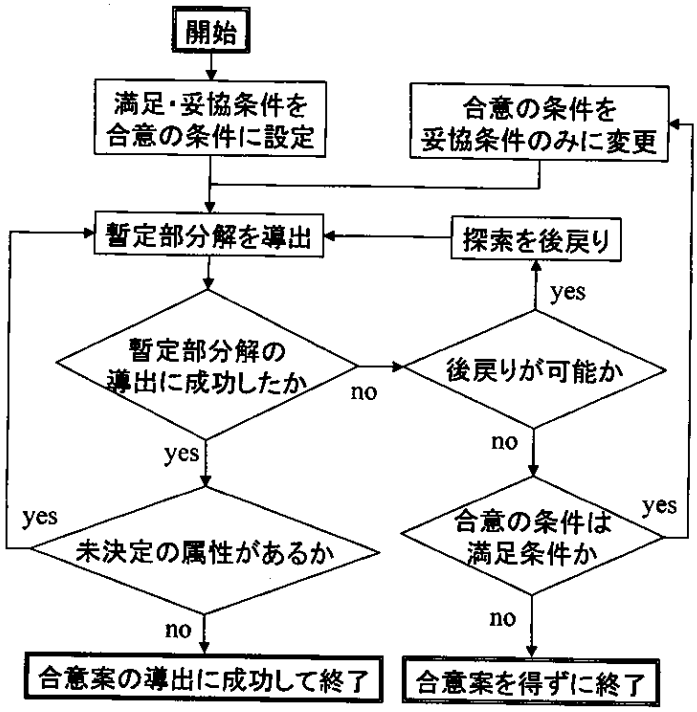

図 3.2: 交渉全体の流れ

部分解に含める属性の追加は，各交渉者がグルーピング知識に定めたグループ単位で行い，部分解を決定する際に交渉者間で調整する属性を原則として新たに追加されたグループの属性に限定することで，スムーズに合意案探索を進める。部分解の拡大時には，一方の交渉者が追加グループの属性に初期値を設定し，初期部分解を作成する。そして初期部分解を交渉者間で調整し，両交渉者の制約条件・合意の条件に違反しない，拡大した部分解を導出する。ただし，制約条件・合意の条件に違反しない新たな部分解を作成できない場合には，探索を後戻り，つまり初期部分解として新たに加えられたグループの属性を再び部分解から除き，最後に調整したグループに属する属性について再調整を行う。このように調整済みの部分解は暫定的なものであり，暫定部分解と呼

ぶ。全ての属性値について両交渉者の制約条件・合意の条件を充足する部分解を作成したとき，それを合意案として交渉を終了する。一方，最初に調整したグループまで探索を後戻りしても暫定部分解を導出できない場合には，その合意の条件下での交渉は決裂となる。

初めは満足条件および妥協条件全ての充足を合意の条件として交渉を行う。その交渉が決裂した場合には，合意の条件を妥協条件のみの充足に変更して再び交渉を行う。このとき，満足条件および妥協条件全ての充足を合意の条件とした交渉において，満足条件には違反するが妥協条件を全て充足するような交渉案（暫定部分解）を作成していた場合には，全属性値が決定していればそれを合意案として交渉を終了，未決定の属性値がある場合にはそれらの属性値の調整から交渉を開始する。妥協条件下での交渉が決裂した場合は，合意案を導出することなく交渉を終了する。

初期部分解の導出について第3.4.3節で，暫定部分解の導出について第3.4.4節で，それぞれ説明する。

3.4.3 初期部分解の導出

第3.4.2節で述べたように，暫定部分解を受け取った交渉者が，グルーピング知識を参照し，属性値が未決定のグループの中から次に調整するグループを任意に1つ選択する。そして自分の制約条件・合意の条件に違反しないように，そのグループの全属性を設定する。この際，暫定部分解に含まれる属性の値は固定しておく。新たなグループの属性値を計算によって決定できない場合は，デフォルトの属性値を用いる。

3.4.4 暫定部分解の導出

暫定部分解とは，いくつかのグループに属する属性について，両交渉者の制約条件・妥協条件に違反しないよう属性値を調整した交渉案である。各交渉者が図3.3に示す流れにしたがう交渉エンジンを持ち，両者間で交渉案をやり取りすることにより，暫定部分解を導出する。

交渉相手が新たな交渉案を伝達してきた場合，その交渉案を評価する。交渉相手が作成した交渉案が自分の制約条件・合意の条件を全て充足する（または，

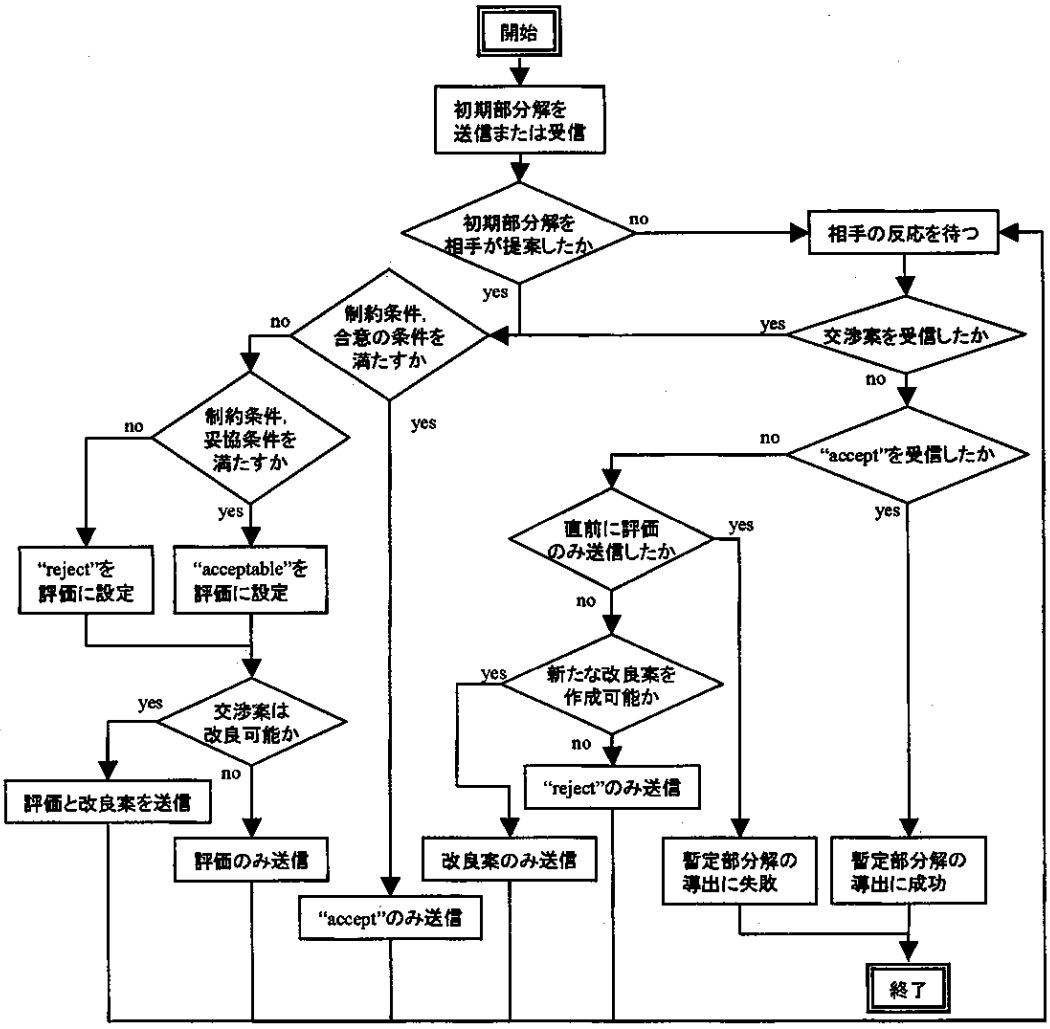

図 3.3: 暫定部分解導出の流れ

その可能性がある）ならば，評価は "accept"（満足）である。制約条件や妥協条件に違反違反している場合は，評価は "reject"（拒否）である。満足条件と妥協条件の充足が合意の条件であり，相手の交渉案が満足条件には違反しているが制約条件・妥協条件は充足しているという場合には，評価は "acceptable"（妥協すれば合意可能）である。"acceptable" と評価された交渉案は，探索を後戻りして合意の条件を妥協条件のみに変更した際の合意案候補となる。そして，交渉相手が作成した交渉案を評価した結果が "accept" でない場合は，第 3.4.5 節で述べるように，目的分析知識を用いて相手の交渉案を改良した交渉案を作成し，評価結果と同時に交渉相手に伝達する。ただし，属性値を変更可能なのは最後に部分解に加えたグループに属する属性だけである。制約条件・

合意の条件に違反しない交渉案を新たに作成できない場合は，直前に受信した交渉案に対する評価のみを伝達し，交渉相手が別の交渉案を作成するのを待つ。相手の交渉案を評価した結果が "accept" であるならば，その交渉案を暫定部分解とする。値が未決定の属性がある場合は，次の初期部分解を作成して "accept" と一緒に交渉相手へ伝達する。全ての属性値が決定した場合は "accept" のみを伝達し，その交渉案を合意案として交渉を終了する。

また，交渉相手が新たな交渉案を伝達してこない場合もある。"accept" のみを受信した場合には，自らが直前に伝達した交渉案を合意案として交渉を終了する。"accept" 以外を受信した場合には，以前に受信した相手の交渉案を改良して新たな交渉案を作成し，交渉相手に伝達する。新たな交渉案を作成できない場合，暫定部分解の導出は失敗である。"reject" のみを交渉相手に伝達して導出失敗を伝え，探索を後戻りする。

3.4.5 交渉案の改良

各交渉者は交渉相手から交渉案を受け取ると，条件充足確認機能により制約条件・妥協条件・満足条件の充足を確認する。条件違反がある場合には，次に示す4つのステップによりその違反を解消する。

(1) 条件充足のための改良属性の選出

条件違反の原因である属性のうち，次式で示す値変更率が最小となる属性を選出する。これにより，交渉相手の制約・妥協・満足条件にも違反しない可能性が大きい改良を行う。ただし，条件充足に必要な変化量は他の属性値を固定した場合を考える。

$$(\text{値変更率}) = \left| \frac{(\text{条件充足に必要な変化量})}{(\text{相手からの交渉案内の属性値})} \right|$$

(2) 目的分析知識の利用

ステップ (1) で選択した属性を改良属性とし，目的分析知識から妥協属性と属性値の変更方法を決定する。

(3) 改良属性の変更幅の決定

条件違反を解消するよう，改良属性の値を条件充足に必要な値に変更する。

(4) 改良後の妥協属性値の決定

- 満足条件充足が合意の条件の場合:
  妥協属性の値を満足条件充足の限界値に変更する。
- 妥協条件充足が合意の条件，かつ，妥協属性が満足条件を充足している場合:
  妥協属性の値を満足条件充足の限界値に変更する。
- 妥協条件充足が合意の条件，かつ，妥協属性が満足条件を充足していない場合:
  第 3.4.1 節で述べたように，妥協属性の値を，妥協条件充足の限界値に徐々に近づけるよう変更する。そのために，妥協属性の値の変更幅を次式のように定義する。ただし，α は，交渉案・改良属性・妥協属性の組み合わせ毎にカウントする妥協回数である。これにより，妥協属性値の妥協条件充足の限界値からの余裕を半減してゆく。

$$(\text{属性値変更幅}) = |\,(\text{現在値}) - (\text{条件充足の限界値})\,| \times (1 - (1/2)^{\alpha})$$

ステップ (3) の条件充足に必要な値，ステップ (4) の条件充足のための限界値などは，属性値逆算機能により求める。ステップ (4) において，同じ妥協属性ばかり変更していると，その属性値は条件充足のための限界値に収束してしまう。そこで，同じ妥協属性ばかり変更しないよう，妥協回数 α には上限値を設定しておく。α が上限に達した場合，別の妥協属性の変更による交渉案の改良を行う。全ての妥協属性を既に α の上限回数まで変更している場合には，改良属性の変更のみによる改良を行う。改良属性のみによる改良を行っても交渉相手が "reject" を伝達してくる場合は，ステップ (1) において改良属性を値変更率が次に小さい属性に変更して改良を行う。変更していない改良属性がない場合，または 1 つの属性の変更では条件違反を解消できない場合には，交渉案の改良は失敗となり，"reject" を伝達する。

1 つの属性の変更では条件違反を解消できないという状況は，交渉案の解空間が細く分布している場合に発生する可能性が大きい。それは，改良前の交渉案を 1 つの軸（属性）の方向にのみ移動させても交渉案の解空間に入りにくいためである。しかし，複数の属性を変更した交渉案を作成した場合には，基に

した相手の交渉案からの変更の度合いが大きく，相手の制約条件に違反する可能性が大きくなると考えられる。このように交渉案の解空間が小さい交渉問題では，人間同士で交渉を行う場合と同様に，本方式を用いても合意案の発見が困難である。

3.5 評価実験

提案手法を次に示す企業間の交換分配交渉問題に適用する。ある会社が製品のサンプルを10万個だけ，一定単価で出荷する。A社とB社はこのサンプルの分配をめぐって交渉する。一方でA社は普段B社が製造するモジュールを購入しているため，サンプルの分配が難航する場合にはそのモジュールを特別に売買することで調整する。

この問題では，普段のモジュールの取引価格により，両者の満足条件を同時に満たす解空間が変化する。両者の満足条件を同時に満たす解空間が存在する場合（問題設定1），および，両者の満足条件を同時に満たす解空間は存在しないが妥協条件を同時に満たす解空間は存在するという場合（問題設定2）のそれぞれについて提案手法を適用し，合意案を発見可能であるか確認する。

3.5.1 問題の構造

この事例で扱うリソースは次のとおりである。

- 共通リソース: $samples = 100000$
- 交渉者Aの個別リソース: $money = 100000$
- 交渉者Bの個別リソース: $modules = 50$

また，この交渉によって値を決定する交渉案の属性は次のとおりである。

- $Sample.A$: 交渉者Aに分配されるリソースsamplesの数量
- $Sample.B$: 交渉者Bに分配されるリソースsamplesの数量
- $Money.AB$: AからBに渡されるリソースmoneyの数量

- $Money.date$: A から B にリソース money が渡される期日
- $Module.BA$: B から A に渡されるリソース modules の数量
- $Module.date$: B から A にリソース modules が渡される期日

それから，制約条件や各交渉者の妥協条件，満足条件を表 3.2 および表 3.3 に示す。交渉者 A の個別制約条件の 1 つに普段のモジュールの価格を表す $price$ があり，$price = 5000$（設定 1）の場合と $price = 4500$（設定 2）の場合について実験を行う。ただし，$price$ 以外の条件は同一である。設定 1 においては，交渉者 A，B 両者の満足条件を同時に満たす解空間が存在する。一方，問題設定 2 においては，両者の妥協条件を同時に満たす解空間は存在するが，満足条件を同時に満たす解空間は存在しない。

表 3.2: 共通制約条件および交渉者 A の個別制約条件・妥協条件・満足条件

共通制約条件	$Sample.A \geq 0$
	$Sample.B \geq 0$
	$Sample.A + Sample.B \leq samples$
交渉者 A の個別制約条件	$price = 5000$（設定 1）， $price = 4500$（設定 2）（普段のモジュールの価格）
	$0 \leq Money.AB \leq money$
	$Money.date \geq 0$
	$Module.BA \geq 0$
	$Module.date \geq 0$
	$Sample.A \geq 30,000$
交渉者 A の妥協条件	$Sample.A \geq 60000$ or $Sample.A + price \times Module.BA - Money.AB \geq 60000$
交渉者 A の満足条件	$Module.date \leq 10$
	$Sample.A \geq 70000$ or $Sample.A + price \times Module.BA - Money.AB \geq 70000$

表 3.3: 交渉者 B の個別制約条件・妥協条件・満足条件

交渉者 B の個別制約条件	$cost = 3000$（モジュールの生産コスト）
	$ability = 20$（モジュールの 1 日の最大生産可能数）
	$Money.AB \geq cost \times Module.BA$
	$Money.date \geq 0$
	$0 \leq Module.BA$ $\leq modules + ability \times Module.date$
	$Module.date$ $\geq \max(0, (Module.BA - modules)/ability)$
	$Sample.B \geq 50000$
交渉者 B の妥協条件	$Sample.B \geq 70000$ or $Sample.B + Money.AB - cost \times Module.BA \geq 70000$
交渉者 B の満足条件	$Sample.B \geq 80000$ or $Sample.B + Money.AB - cost \times Module.BA \geq 80000$

3.5.2 使用する知識

表 3.4 に両交渉者の属性のグルーピング知識およびデフォルトの属性値を，また，表 3.5 に両交渉者の目的分析知識をそれぞれ示す。

表 3.4: グルーピング知識とデフォルトの属性値

属性	交渉者 A		交渉者 B	
	グループ	デフォルト値	グループ	デフォルト値
$Sample.A$	1	70000	1	20000
$Sample.B$	1	30000	1	80000
$Money.AB$	2	0	2	0
$Money.date$	3	0	3	0
$Module.BA$	2	0	2	0
$Module.date$	3	0	3	0

表 3.5: 目的分析知識

交渉者 A		交渉者 B	
改良属性	妥協属性	改良属性	妥協属性
$Sample.A$ +	$Module.BA$ − $Money.AB$ + $Module.date$ +	$Sample.B$ +	$Module.BA$ + $Money.AB$ − $Module.date$ −
$Money.AB$ −	$Sample.A$ − $Module.BA$ − $Module.date$ +	$Money.AB$ +	$Sample.B$ − $Module.date$ −
$Module.BA$ +	$Sample.A$ − $Money.AB$ + $Module.date$ +		

3.5.3 問題設定 1 の交渉経過と結果

本事例の問題設定 1 に対して提案手法を適用したときの交渉の過程と結果を表 3.6 に示す。表 3.6 の各行は，交渉相手が直前に作成した交渉案に対する評価と，新たに作成して相手に送信する交渉案の属性値を示している。ただし，各行において交渉者間で調整中のグループ以外の属性値は省略している。交渉者 A がグループ 1 に属する $Sample.A$ と $Sample.B$ に属性値を設定し，7 万個のサンプルを要求することで交渉を開始する（表 3.6 中の#1）。

表 3.6 中の#3 において交渉者 A が作成した交渉案は，交渉者 B の合意の条件（制約条件，妥協条件，満足条件）を全て充足する可能性がある。そのため，#3 に対する交渉者 B の評価は "accept" であり，#4 以降では $Sample.A$ と $Sample.B$ の値を $\{Sample.A$=45000, $Sample.B$=55000$\}$ に固定して，グループ 2 に属する $\{Money.AB,\ Module.BA\}$ の属性値を調整してゆく。

交渉案改良の一例として，交渉者 B が表 3.6 中の#5 の交渉案を受けてから#6 の交渉案を作成する過程を以下で説明する。交渉者 A が提案した#5 の交渉案は，交渉者 B の妥協条件に違反している。このとき調整しているグループに属する属性は $Money.AB$ と $Module.BA$ であり，この違反を解消し，かつ

表 3.6: 問題設定 1 における交渉経過

#	提案者	評価	交渉案
1	A	—	*Sample.A*=70000, *Sample.B*=30000
2	B	reject	*Sample.A*=20000, *Sample.B*=80000
3	A	reject	*Sample.A*=45000, *Sample.B*=55000
4	B	accept	*Money.AB*=25000, *Module.BA*=0
5	A	reject	*Money.AB*=25000, *Module.BA*=10
6	B	reject	*Money.AB*=55000, *Module.BA*=10
7	A	reject	*Money.AB*=55000, *Module.BA*=16
8	B	reject	*Money.AB*=73000, *Module.BA*=16
9	A	reject	*Money.AB*=73000, *Module.BA*=20
10	B	reject	*Money.AB*=85000, *Module.BA*=20
11	A	acceptable	*Money.AB*=85000, *Module.BA*=22
12	B	acceptable	*Money.AB*=91000, *Module.BA*=22
13	A	acceptable	*Money.AB*=91000, *Module.BA*=24
14	B	acceptable	*Money.AB*=97000, *Module.BA*=24
15	A	acceptable	*Money.AB*=97000, *Module.BA*=25
16	B	acceptable	*Money.AB*=100000, *Module.BA*=25
17	A	accept	*Money.date*=0, *Module.date*=0
18	B	accept	

満足条件を全て充足するときの属性値変更率は,

- $Money.AB$: $|+30000/25000| = 1.2$
- $Module.BA$: $|-10/10| = 1.0$

である。したがって，まずは $Module.BA$ を減少させて交渉案の改良を試みる。目的分析知識を参照すると，“$Module.BA : -$” に対応する妥協属性は存在しないので，妥協属性を用いないで $Module.BA$ だけを変更して改良を試みる。しかし，$Module.BA$ を満足条件充足まで減少させただけの交渉案は，表 3.6 中の#4 において同じ交渉案が既に提案されており，また変更幅を半減すると

満足条件を充足できなくなる。そこで，次に $Money.AB$ を増加させて交渉案を改良を試みる。“$Money.AB$: +” に対応する妥協属性は同じグループに含まれない属性ばかりであるため，先程と同様に要求属性だけを変更しただけの，表 3.6 中の#6 の交渉案を作成する。

このようなプロセスでお互いに交渉案の改良を繰り返すことにより，全ての属性に属性値が設定された交渉案#17 を交渉者 A が提案したのに対して交渉者 B が “accept” のみを送信し，交渉は終了した。こうして得られた合意案を表 3.7 に示す。この合意案は，確かに両交渉者の制約条件，妥協条件，満足条件を全て満たしている。

表 3.7: 問題設定 1 において導出された合意案

属性	属性値
$Sample.A$	45000
$Sample.B$	55000
$Money.AB$	100000
$Money.date$	0
$Module.BA$	25
$Module.date$	0

3.5.4 問題設定 2 の交渉経過と結果

問題設定 2 は，交渉者 A と B の満足条件を同時に満たす解空間は存在しないが妥協条件を同時に満たす解空間は存在するという設定である。この問題に提案手法を適用したときの交渉の過程の一部を表 3.8 に示す。

表 3.8 中の#36 において，交渉者 B が “reject” のみを送信している。これは，$\{Money.AB,\ Module.BA\}$ のグループについて，交渉の経過を後戻りしながら#35，#33，#31 といった交渉者 A が作成した交渉案全てに対し第 3.4.5 節で述べた提案手法による交渉案の改善を試みても，既に提案済みである交渉案しか作成できないためである。#36 の “reject” を受けて，今度は交渉者 A が交渉の経過を後戻りしながら新たな交渉案の作成を試み，#30 の交渉案を基にした新たな交渉案#37 を提案している。

表 3.8: 問題設定 2 における交渉経過（一部抜粋）

#	提案者	評価	交渉案
...		...	
3	A	reject	*Sample.A*=45000, *Sample.B*=55000
4	B	accept	*Money.AB*=25000, *Module.BA*=0
...		...	
13	A	reject	*Money.AB*=100000, *Module.BA*=28
14	B	acceptable	*Money.AB*=109000, *Module.BA*=28
...		...	
30	B	reject	*Money.AB*=73000, *Module.BA*=16
31	A	reject	*Money.AB*=47000, *Module.BA*=16
32	B	reject	*Money.AB*=47000, *Module.BA*=7
33	A	reject	*Money.AB*=6500, *Module.BA*=7
34	B	reject	*Money.AB*=46000, *Module.BA*=7
35	A	reject	*Money.AB*=46000, *Module.BA*=16
36	B	reject	—
37	A	—	*Money.AB*=73000, *Module.BA*=22
38	B	reject	*Money.AB*=91000, *Module.BA*=22
...		...	
44	B	reject	*Money.AB*=74000, *Module.BA*=16
45	A	reject	—
46	B	reject	—
47	A	—	*Sample.A*=57500, *Sample.B*=42500
48	B	accept	*Money.AB*=37500, *Module.BA*=0
...		...	

表 3.8 中の#45，#46 においては，両交渉者が "reject" のみを送受信している。これは $\{Sample.A=45000,\ Sample.B=55000\}$ の下での $\{Money.AB,\ Module.BA\}$ のグループの暫定部分解導出に失敗したことを表しており，#47 において $\{Sample.A,\ Sample.B\}$ についての暫定部分解の導出に後戻りして

いる。

#49 以降にも探索の後戻りが繰り返し発生し，およそ 800 回の交渉案のやり取りの後，$\{Sample.A,\ Sample.B\}$ についての新たな暫定部分解を導出できなくなったため，合意の条件から満足条件を除外した。このとき，交渉者 A が作成した#13 の交渉案に対する交渉者 B の評価が "acceptable"（妥協すれば合意可能）であったため，これを基にした交渉案，すなわち，#13 に最後の属性グループ $\{Money.date=0,\ Module.date=0\}$ を加えた交渉案を交渉者 B が作成した。これは交渉者 A の合意の条件を満たすので，交渉者 A は "accept" のみを交渉者 B に送信した。こうして，妥協条件のみを合意の条件として以降は，2 回のやり取りによって表 3.9 に示す合意案を導出した。

表 3.9: 問題設定 2 において導出された合意案

属性	属性値
$Sample.A$	45000
$Sample.B$	55000
$Money.AB$	100000
$Money.date$	0
$Module.BA$	28
$Module.date$	0

3.5.5 考察

分配するサンプル数は両交渉者の希望数の合計より少なく，また，部品生産能力・コストなどの交渉者固有の制約条件が存在するなど，一見合意が難しそうな問題設定だった。しかし，問題設定 1，2 両方において合意案を導出することができた。問題設定 1 においては，導出された合意案は両交渉者の満足条件を全て充足しており，A 社は B 社からモジュールを安く購入することができ，一方 B 社は，利益がでる価格でモジュールの在庫を捌くことができた。つまり，両者にとって十分に高評価の解を得られたと言える。

一方，問題設定 2 は，両交渉者の満足条件を同時に満たす解空間は存在しないが妥協条件を同時に満たす解空間は存在するという問題であった。導出され

た合意案は交渉者Aのみの満足条件を充足しており，交渉者Bのみが妥協した形である。また，交渉案の属性の数が6であるのに対し，合意の条件から満足条件を除外するまでに800回以上の交渉案のやり取りが必要であった。提案手法の適用先として想定している実際の受発注においては，より多くの交渉案の属性が使用されると考えられる。したがって，より効果的な交渉案を作成する方法や，両交渉者の満足条件を同時に充足する解空間の存在を早い段階で判定する方法などの考案が今後の課題として挙げられる。

また，評価実験に用いた事例では，共通リソースが1種類，個別リソースが各交渉者に1種類ずつであった。一般に，受発注交渉における発注者は個別リソースとしてお金を持ち，その他の全てのリソースはお金に換算可能である。この事例においても，サンプルの配分を表す $Sample.A$ や $Sample.B$，モジュールの取引量を表す $Module.BA$ と，お金の取引量を表す $Money.AB$ とが同時に含まれるような条件式を，両交渉者とも満足条件や妥協条件に持っている。また，提案手法において，共通リソースは元の所有者が交渉者以外の第三者であること以外には個別リソースとの本質的な違いがない。したがって，共通リソースが存在しないような問題や，受注者の販売製品を表す個別リソースが複数存在するような問題に提案手法を適用した場合にも，本事例と同様の交渉案を得ることができる。

3.6 結言

本章では，二者間交換分配交渉に対して，評価関数の代わりに満足条件，妥協条件と呼ぶ複数の条件式を用い，制約充足問題として問題のモデル化を行った。そして，各交渉者のエージェントとなる計算機同士で二者間交換分配交渉を自動的に行うための手法として，交渉者がお互いに相手が作成した交渉案を改善しながら交換する交渉方式を提案した。本手法では，交渉案を自らの条件を充足するよう改善する際に目的分析知識と呼ぶ簡単な知識を用いることにより，交渉相手の合意の条件を充足する可能性が大きい交渉案を作成する。また，交渉開始時には合意の条件として妥協条件だけでなく満足条件も充足する交渉案，すなわち両交渉者にとって高評価となる合意案の作成を試みる。そのような合意案が得られない場合には，合意の条件から満足条件を除外することによって探索範囲を拡大し，合意案導出の可能性を高める。

提案方式を企業間の受発注時における交渉問題に適用した。両交渉者の満足条件を同時に満たす解空間が存在する問題に対し，その解空間内で合意案を導出可能であることを確認した。一方，そのような解空間が存在しない問題に対し，合意の条件から満足条件が除外されるまでに交渉者間で多数のやり取りが必要なことがわかった。そのため，提案システムを実装して多様な問題を解くことにより，問題の性質と探索時間との関係の確認や，提案手法によって作成される合意案が両交渉者にとって妥当な内容であるかどうかの検証を行い，探索を効率化することが今後の課題である。

また，実際の受発注業務にて使用することを考えた場合には，EDI などのシステムと連携するためのインタフェースの設計や，特に合意の条件から満足条件が除外された後に導出された合意案について，妥当な案が得られているか確認することなども課題となる。

第4章

グループ制約付き大規模ジョブショップ問題向きの確率的ルール適用方式による統合タブーサーチシステム

4.1 緒言

各企業における製品の生産性が向上すれば，低価格化や納期の短縮により下流企業ではその製品を入手しやすくなり，サプライチェーン（SC）全体の効率化につながる可能性がある。SCを流れる製品やその製造方法はさまざまであり，それぞれに固有の制約や評価基準が存在するため，生産性向上のためにはその対象に即した対策が必要となる。本章では，グループ制約という特殊な制約を持つ大規模なジョブショップ問題（JSP）を対象とする。

JSPとは生産分野における計画問題の一種であり，いくつかの処理が必要な複数のジョブを，ラインと呼ばれるいずれかの処理が可能な複数の機械資源に割り当てる問題である。処理を行う順序によって手間や設備の負担が大きく異なるが，この問題はNP困難であり計算機で最適解を求めることが非常に難しいため，熟練した専門家が独自のノウハウにより手作業で計画を作成している場合が多い。そのノウハウの1つがグループ制約であり，同一の処理工程を必要とするジョブはグループにまとめてラインに割り当てるというものである。

ある企業は，ラインに割り当てるべきジョブの数が約2000のグループ制約を持つJSPに対し，現状では専門家の手作業により8時間かけて計画を作成

しているが，これと同等の計画を2時間以内に作成したいという要求を持っている。これに対し，大前らは専門家の手作業による計画と評価値で比較して95%の計画を2時間で作成可能な，確率的ルール適用方式を用いたタブーサーチ手法を提案している [72, 73]。しかし，計画の評価として，専門家は感覚的に判断しているが定式化が困難な評価項目が存在するため，大前らの手法によって作成した計画を，専門家の判断に基づいて修正することが必要である。そこで，大前らの手法を並列化することにより従来と同等の計画を1時間程度で作成し，残る1時間でその計画を対話的に修正する手法を提案する。また，季節によるジョブの傾向の変動や生産設備の変更などに伴い，スケジューリングに用いるパラメータの適切な値も変化する。このパラメータを自動的に調整する手法を提案する。これらの機能を統合したスケジューリングシステムを開発する。

以下，第4.2節では，本章で対象とするグループ制約を持つJSPについて説明する。第4.3節では提案システムを構成する各モジュールについて説明する。第4.4節では評価実験を行い，提案システムの有効性を示す。

4.2 グループ制約付きJSP

4.2.1 問題概要

グループ制約付きJSPは，同じ処理工程を必要とする複数のジョブをグループ化して，そのグループ単位でラインに処理時間の割当を行う計画問題である。一般的なJSPは，図4.1に示すように，ジョブと呼ばれる製品製造に必要な複数の処理工程の集合を，ラインと呼ばれる処理を実行する複数の機械資源に割り当てる問題である。問題の制約として，ジョブ毎に製品になるまでに必要な処理工程とその順番が，また，ライン毎に実行可能な処理が，それぞれ定まっている。また，現実の生産現場においてはラインの休止・再起動や処理の切り替えを行うのに大きなコストが必要となる。そのため，同じ処理工程を必要とするジョブが複数存在する場合にはそれらをひとまとめにして処理する方が効率がよいので，そのように処理を行うよう制約を設ける場合がある。これをグループ制約と呼ぶ。例えば，図4.1に示した計画にグループ制約を適用すると，ラインB，C，Dにおいて処理b2，c2，d1のジョブをそれぞれ1つのグ

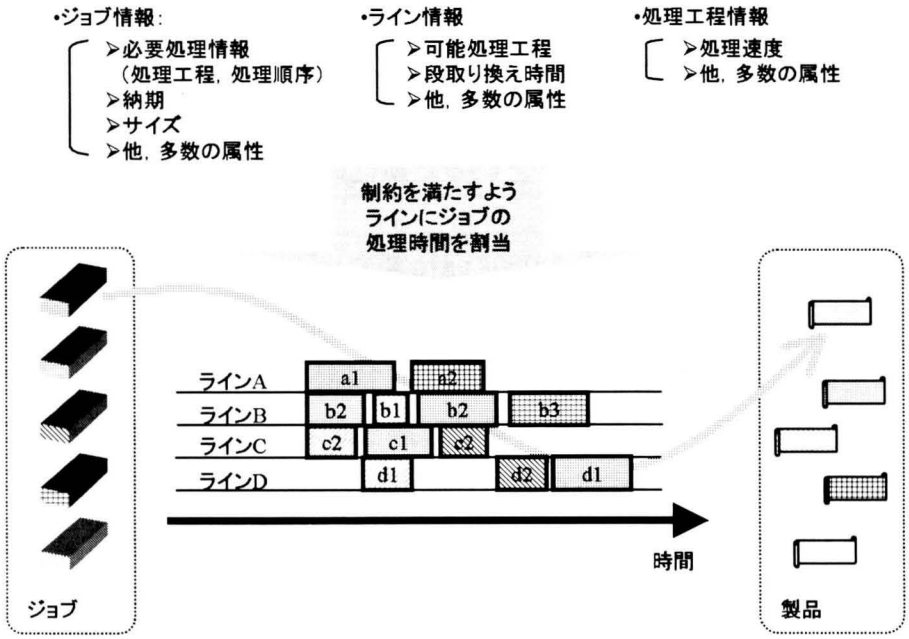

図 4.1: ジョブショップ問題の概要

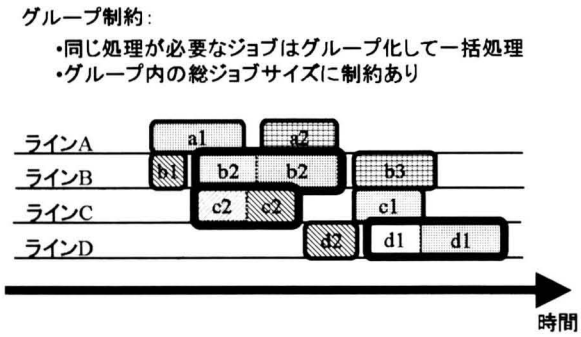

図 4.2: グループ制約の適用

ループにまとめて処理するため，図 4.2 のような計画になる。

すなわち，問題は「ジョブ情報」，「処理工程情報」，「ライン情報」からなり，ジョブのグループをラインに割り当てた計画が解となる。「ジョブ情報」には，各ジョブのサイズ，納期，製品となるまでに必要な処理工程の処理コードやその順序などの属性値が存在する。「処理工程情報」には，処理コードをキーとして，各処理工程を行うグループに対するグループサイズの制約（最大サイズ，最小サイズ），単位サイズあたりの処理時間，処理後に次工程の開始まで空けなければならない時間などの属性値が存在する。「ライン情報」としては，各

ラインで実行可能な処理コード，ライン間をジョブが搬送されるのに要する時間，搬送されてきたジョブが処理開始可能になるまでに要する時間，処理するグループを切り替える際に発生する段取り換え時間や，処理休止の後に再度立ち上げるための時間などのコストといった属性値が存在する。

4.2.2 計画の評価

計画の定量的な評価項目として，平均製造工期，グループ切替回数，ライン休止回数の3つを用いる。ここで，平均製造工期とは，製造工期（あるジョブの先頭工程が処理されるグループの処理開始から最終工程が処理されるグループの処理終了までの時間）の全ジョブでの平均値である。グループ切替回数とは，各ラインで処理するグループの処理コードが変わる回数である。ライン休止回数とは，各ラインでの休止時間（何も処理が割当てられていない時間帯）が発生する回数である。

製造工期を短縮することと，グループ切替回数や休止回数を減少させることとの間にはトレードオフの関係が存在し，得られる計画を評価する際には総合的な判断が必要となるため，3つの評価指標に重み付けを行い，その和を計画の評価関数として用いる。この評価関数をVで表し，平均製造工期Tの重みをW_T，グループ切替回数Cの重みをW_C，ライン休止回数Rの重みをW_Rとして，以下の式のように定義する。ただし，C，Rは，それぞれグループ切替回数，休止回数をライン毎に重み付けして足した値である。

$$V = W_T \times T + W_C \times C + W_R \times R \tag{4.1}$$

本問題では，この評価関数Vの値が最小となる計画を作成することが目的となる。ただし，評価関数Vに含まれる評価項目以外にも，専門家の手作業による計画に近づけるための評価指標や，定量化が困難な評価指標が存在する。そのため，計画の最終的な評価は専門家の手にゆだねる必要がある。

4.3 確率的ルール適用方式による統合タブーサーチシステム

4.3.1 提案システムの構成

本章で対象とするJSPはジョブ数が約2000と大規模であるため，最適解を求めるのは非常に困難である。また，計画を感覚的・経験的に評価する専門家の判断にしたがって解を修正しなければならないので，要求の入力とそれを満たす解の探索の試行錯誤的な繰り返しを高速に行うことが必要である。そこで，図4.3に示すような構成の統合タブーサーチシステムを提案する。

タブーサーチとは，組み合わせ最適化問題に対する汎用の逐次改善法である[70]。タブーサーチによってi番目の暫定解x_iから（$i+1$）番目の暫定解x_{i+1}に遷移する様子を図4.4に示す。まず，暫定解x_iに僅かに変更を加えた解の集合である近傍集合$N(x_i)$を作成する。次に，$N(x_i)$の中で最良の解yを選択し，

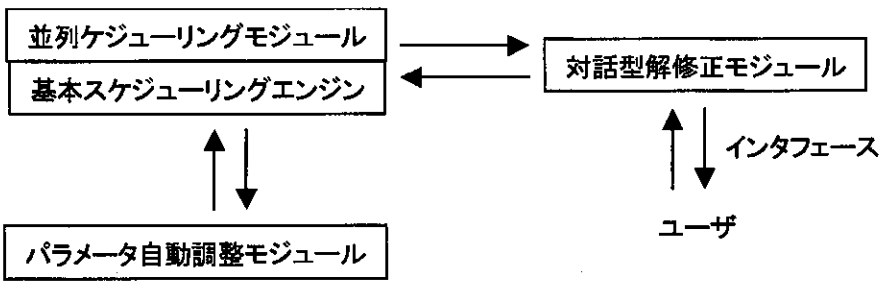

図 4.3: 統合タブーサーチシステムの構成

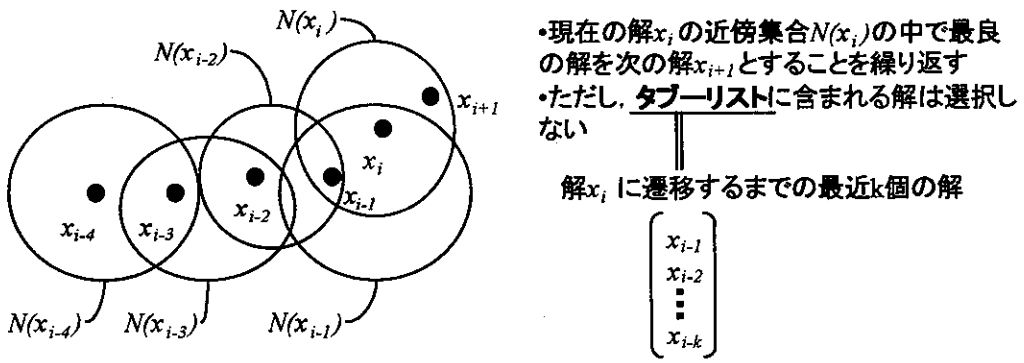

図 4.4: タブーサーチの概要

たとえ y が x_i より劣る場合でも，それを新たな暫定解 x_{i+1} とする。この操作を繰り返すことにより解を改良するが，その際に同じ解が再び暫定解に選ばれることがないよう，“タブーリスト” を用いる。タブーリストとは最近 k 個の解の集合であり，タブーリストが保持する解の数 k をタブーリストのサイズと呼ぶ。近傍集合 $N(x_i)$ の作成において，タブーリストに含まれる解は $N(x_i)$ から除外する。

提案システムの核となるのは，確率的ルール適用方式を用いたタブーサーチによる既開発の基本スケジューリングエンジン [72, 73] である。このエンジンでは，まずヒューリスティクスによりジョブの自由度に基づいて初期解を1つ作成し，その解をタブーサーチによって改善する。このタブーサーチにおいて解を改善，すなわち暫定解の近傍集合を作成する際には，経験に基づいて作成された複数のディスパッチングルールを，各ルールに設定された適用確率にしたがってランダムに適用する。

しかし，対象とする問題があまりに大規模であるため，基本スケジューリングエンジンのみでは計算速度が不十分である。そのため，基本スケジューリングエンジンに漸近探索方式および探索フェーズ分割方式を導入し，また，並列スケジューリングモジュールを導入することにより探索を高速化する。

また，スケジューリングエンジンではさまざまなパラメータが使用される。その中でも，ルール適用確率は解の質に大きな影響を与え，また，探索フェーズを移行すべきタイミングは問題によって異なり，これらを人の手によって適切に調整するには膨大な時間を要する。そこで，これらパラメータの自動調整モジュールを導入する。

最後に，評価関数に取り入れることが困難な評価指標に関して専門家の要求に応じて解を修正するために，ガントチャートで計画を確認しながら修正要求を繰り返し入力可能な対話型解修正モジュールを導入する。対話型解修正モジュールにおいては，計画確認と要求入力のためのインタフェース，および高速に修正するための，タブーサーチのルール適用確率制御手法を提案する。

4.3.2 基本スケジューリングエンジン

基本スケジューリングエンジンでは，次に示す2つのステップにより解を作成する [72, 73]。

(1) ジョブ自由度分析に基づく初期解の作成

(2) 確率的ルール適用方式を用いたタブーサーチによる解の改善

4.3.2.1 ジョブ自由度分析に基づく初期解の作成

与えられたジョブのデータを分析し，初期解を高速に作成する。ジョブが計画に割り当てられるとき，各ジョブはボトルネックとして計画全体に影響を及ぼす。その影響の度合いをジョブの自由度と定義し，計画全体に与える影響が小さいほど自由度が大きいとする。自由度は次の2つの要素からなる。

- 必要処理工程数
  ジョブによって必要処理工程は異なる。必要処理工程の多いジョブはその分多くのラインを経由するため，計画全体に与える影響が大きい。

- 稼動率
  全ジョブの処理が割当ったと仮定した際，各ラインにおける全処理に必要な時間を稼働時間と呼び，全ライン中で最も長い稼働時間に対する各ラインの稼働時間の比率を“稼働率”と定義する。稼働率の大きいラインは多くのジョブを処理しているので，そのラインの計画の変更は他のラインに大きな影響を与える。

ジョブiの必要処理工程数をL_i，ジョブiのl番目の工程がラインX_lで処理されるとし，ラインX_lの稼働率をO_{X_l}，必要処理工程数の重みをW_{f1}，稼働率の重みをW_{f2}として，ジョブiの自由度を以下の式で定義する。

$$\text{ジョブ}\,i\,\text{の自由度} = -(W_{f1} \times L_i + W_{f2} \times \sum_{l=1}^{L_i} O_{X_l}) \tag{4.2}$$

4.3.2.2 確率的ルール適用方式タブーサーチによる解の改善

タブーサーチを本問題へ適用する際には，評価関数に含まれる3種類の評価項目に着目し，各評価項目それぞれについて改善が見込めるディスパッチングルールを用意する。ルールによって，それを1回適用することによる評価値の改善幅の期待値は異なる。そのため，近傍解を生成する際に全ルールを適用し

た場合，特定のルールによって生成された解ばかりが新たな暫定解として選択される可能性が大きい。つまり，特定の評価項目についてのみ改善され，他の評価項目については改善されない，あるいは悪化した解になる。しかし，特定の評価項目のみが極端な評価である解は，それ以降のタブーサーチにおいて改善されにくかったり，専門家による評価が低かったりする傾向がある。

そこで，近傍集合を生成する際には，それらの中から確率的に選択したルールを暫定解に適用する。これにより，トレードオフの関係にある各評価項目を全体的に改善する。

さらに，漸近探索タブーサーチ方式と探索フェーズ分割方式を導入し，得られる解の評価と探索効率の改善を図る。

漸近探索タブーサーチ　(4.1) 式において，平均製造工期に関する項 T を減少させるルールが適用されると，グループ切替回数に関する項 C が増加する場合が多い。一方，C を減少させるルールは評価値 V の改善効果が小さい。このため，T が小さくかつ C が大きくなると，それ以降 V が改善されにくくなる。

この問題に対し，T を減少させるルールの適用確率を小さくすることが考えられる。しかし，その場合，V の改善効果が小さいルールを多く適用することになるため，解を十分に改善できない可能性がある。

そこで，適用するルールを制限するのではなく，近傍集合から新たな暫定解を選択する際に，C が大きくなり過ぎる解を選択しないよう制限する。すなわち，全ラインでのグループ切替回数の増加がある閾値 γ を超える解は選択しないようにする。短期的には評価値の優れた解をあえて選択しない場合が発生するが，長期的に見るとより優れた解を発見することから，この方式を漸近探索タブーサーチと名付ける。

探索フェーズ分割方式　タブーサーチの序盤と終盤では，解の特徴が大きく異なり，各ディスパッチングルールによる解の改善の効果も異なってくる。そこで，タブーサーチによる探索を複数のフェーズに分割し，フェーズ毎にその探索の進度に適したルールの適用確率を用いることにより，探索全体を通して効率的な解の改善を可能にする。

探索フェーズ分割方式では，どのようなタイミングでフェーズを移行するかが問題となる。初期解の評価値や最終的に得られる解の評価値は入力される

ジョブデータによって異なるため，評価値の絶対値を探索フェーズ移行の基準にすることはできない。そこで，同じ評価関数を用いていれば，評価値の改善量の平均値は探索の進度によって似通った値になると仮定し，評価値改善量の平均値および解の遷移の回数を探索フェーズ移行の基準とする。すなわち，第 q フェーズから第（$q+1$）フェーズへの移行の基準として，評価値改善量の平均値に対する閾値 MT_q，解の遷移の回数に対する閾値 IT_q を設定しておく。そして，第 q フェーズにおいてタブーサーチの実行中，評価値改善量の移動平均が MT_q を下回り，かつそのフェーズ内での解の遷移回数が IT_q を超えたときに，第（$q+1$）フェーズへと移行する。

MT_q および IT_q の値は，第4.3.4節で述べるパラメータ自動調整モジュールによって決定する。

4.3.3 並列スケジューリングモジュール

スケジューリングの高速化のために，並列スケジューリングモジュールを導入し，複数の計算機で並列して計算を行う。第4.3.2.2節で述べたようにタブーサーチは解の逐次改善を繰り返すという手法であるため，得られる解は初期解によって異なる。そこで，異なる複数の初期解に対して並列にタブーサーチを短期間実行することによって優れた解が存在する可能性が高い解空間の領域を見きわめ，その後，その近辺を並列に探索する。

(4.2) 式で示した従来のジョブ自由度に基づく初期解生成手法では，1つしか初期解を生成できない。そこで，ジョブ自由度の要素として納期余裕を導入し，納期余裕を重視する度合いが異なる複数の初期解を生成する。納期余裕とは，ジョブの納期までの時間から，そのジョブの必要処理工程全てに要する時間を引いた時間である。ジョブ i の納期余裕を D_i，納期余裕の重みを W_{f3} とおき，新しいジョブ自由度を次式で定義する。

$$\text{ジョブ}\,i\,\text{の自由度} = -(W_{f1} \times L_i + W_{f2} \times \sum_{l=1}^{L_i} O_{X_l}) + \frac{2S+j}{2S} W_{f3} \times D_i$$
$$(j = 1-S, \cdots, -1, 0, 1, \cdots, S) \quad (4.3)$$

j の値を $(1-S)$ から S まで変化させることによって $2S$ 通りのジョブ自由度を算出し，それぞれに基づいた初期解を生成する。

本並列化方式では，複数のスレーブ計算機がそれぞれ異なる探索空間において解を探索することにより，広範囲な探索空間から優れた解を短時間で発見する。スレーブ計算機の増設・削減を容易に可能とするために，問題入力やスレーブ間での解の比較などの通信は全て1台のマスタ計算機を介して行う。マスタが問題データの入力を受けると，図4.5に示すようにして解を探索する。

(1) マスタが式(4.3)により算出したジョブ自由度に基づいてスレーブと同数以上の初期解を作成し，各スレーブにそれぞれ異なる初期解を問題データとともに送信

(2) 各スレーブにおいて，マスタから受信した解を確率的ルール適用方式を

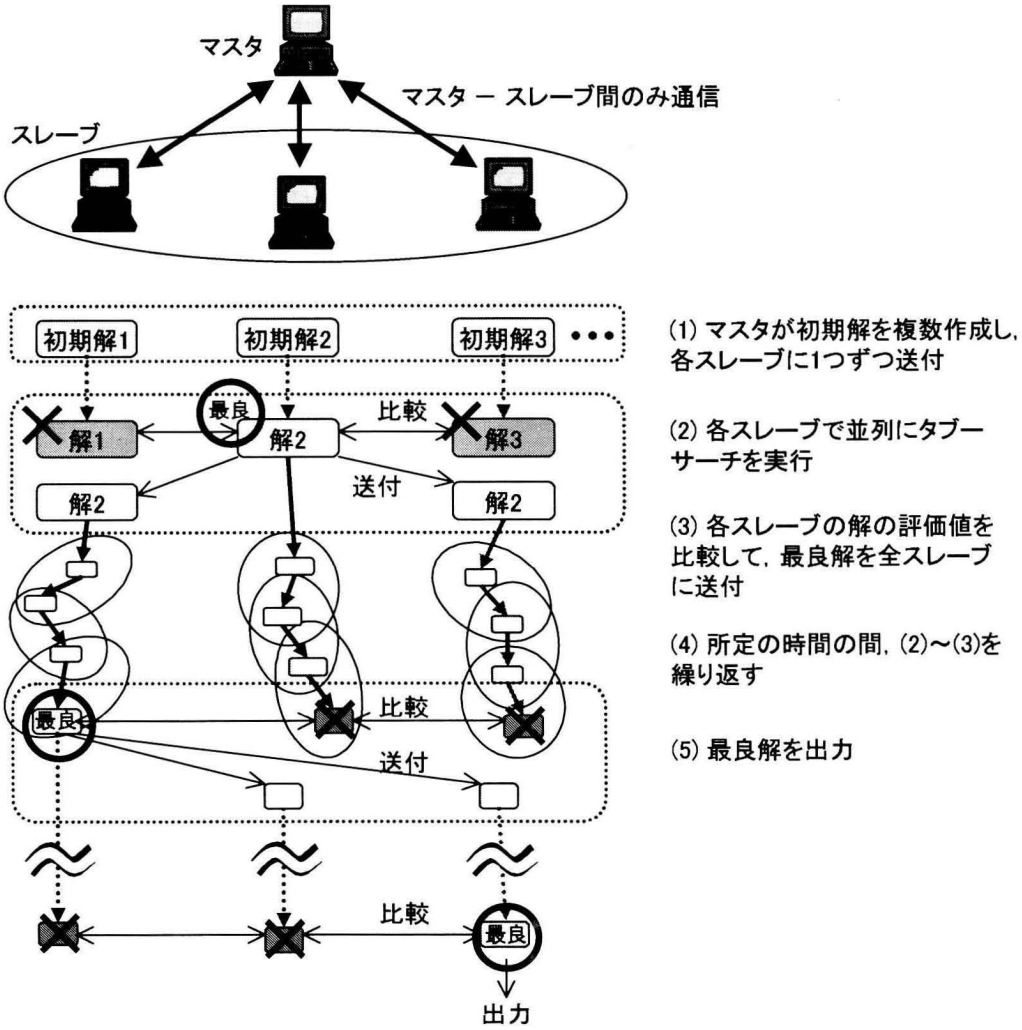

図 4.5: 並列スケジューリングの概要

用いたタブーサーチによって所定の回数 E 回改善した後，改善された解の評価値をマスタに送信

(3) マスタが全スレーブの解の中で最良の解を選択し，その解とそれが作成されたときのタブーリストを，マスタが中継して全スレーブに送信

(4) (2) 以降を所定の期間繰り返した後，最良の解を出力

2 回目以降の (2) の実行においては，全てのスレーブが同じ解を新たな暫定解として確率的ルール適用方式を用いたタブーサーチを行う。このとき，複数のスレーブが同一の解を作成したのでは効率が悪い。そこで，スレーブ毎にそれぞれ異なる擬似乱数系列を使用することにより，解の改善のために適用されるルールがスレーブ毎に異なるようにし，探索を切り分ける。また，その解はいずれかのスレーブ A でタブーサーチによって作成された解であり，それが作成されるまでにスレーブ A が経た暫定解を他のスレーブが再度探索するのは非効率的である。そこで，各スレーブはマスタから解と同時にその解が作成されたときのタブーリストを受け取り，一時的にタブーリストのサイズを k_{EX} ($> k$) に拡大して格納する。タブーリストのサイズは，解の改善を k_{EX} 回繰り返した後に k に戻す。ただし，k は通常時のサイズである。k_{EX} の値は，k_{EX} 回の解改善後の近傍集合にスレーブ A が既に探索した解が含まれなくなるように設定する。

図 4.6 は，スレーブ A が作成した解 x_{10}^A をスレーブ B の新たな暫定解としたときの探索の様子を表している。スレーブ A とスレーブ B では使用する擬似乱数系列が異なるので，作成する近傍集合もまたスレーブ間で異なり，別々の方向へと探索が進んでゆく。また，スレーブ B のタブーリストを一時的に拡大して解 x_{10}^A が作成される直前 5 個の暫定解 $\left(x_9^A, x_8^A, x_7^A, x_6^A, x_5^A\right)$ を格納したので，スレーブ B が探索する空間はスレーブ A が探索してきた経路から離れてゆく。こうして，スレーブ A とスレーブ B の探索空間が分けられる。

4.3.4 パラメータ自動調整モジュール

確率的ルール適用方式を用いたタブーサーチにおいて，各ルールの適用確率は非常に重要なパラメータであり，解の質および計算時間に大きく影響する。

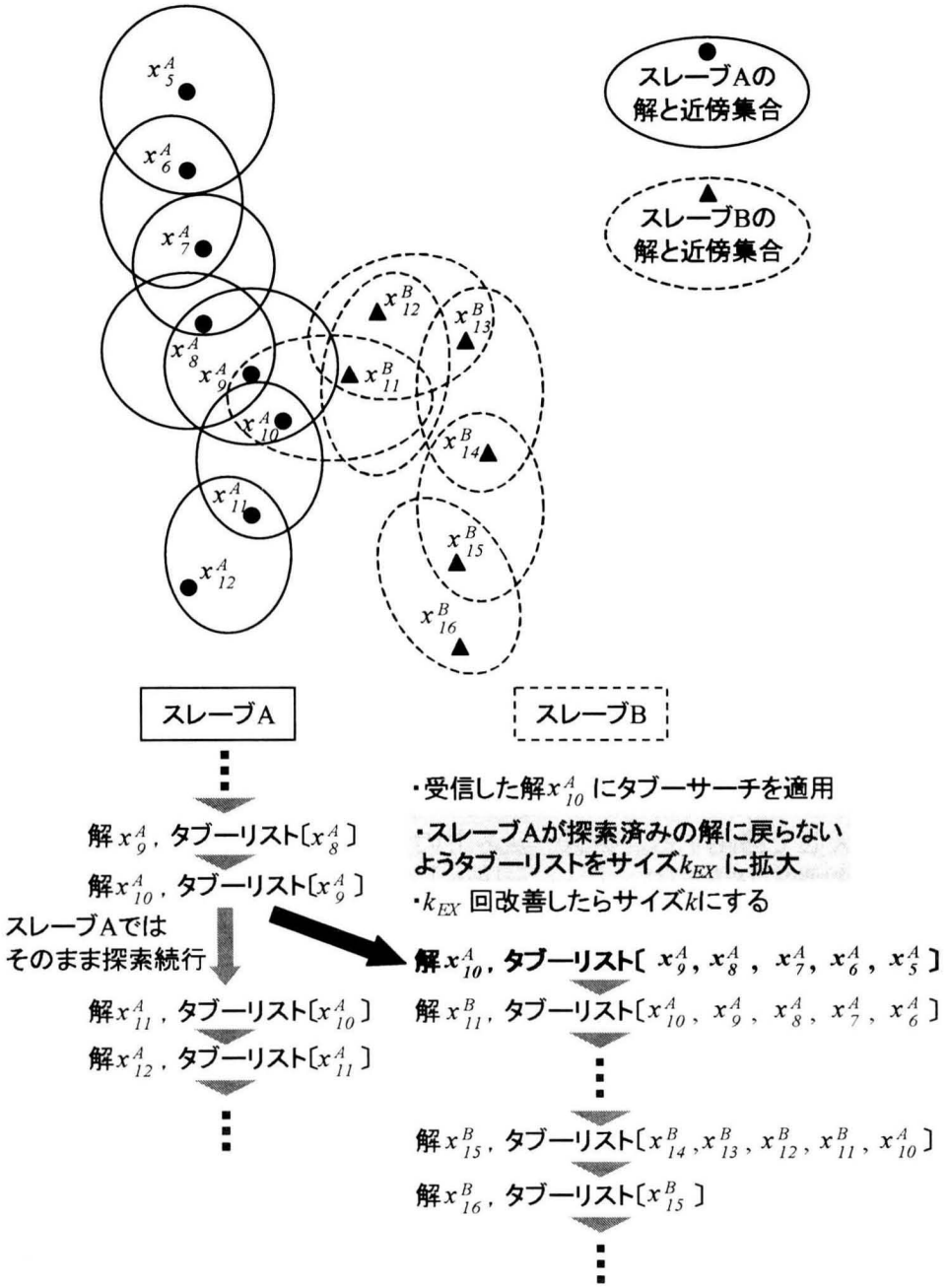

図 4.6: 探索の切り分け ($k = 1$，$k_{EX} = 5$ の場合)

それに加え，これらを手作業で調整するためには専門家の労力と膨大な時間が必要である。また，探索のフェーズを移行する際に基準とすべきパラメータ MT_q，IT_q（q はフェーズ番号）の値は問題によって異なる。そこで，ルール適用確率および探索のフェーズ移行の基準値を自動調整するためのモジュール

を開発する。

このモジュールは，サンプル問題としてラインおよびジョブのデータ，探索のフェーズ分割数 F および各フェーズの境界となる探索の進度 Prg_q ($1 \leq q \leq F-1$) を入力とする。ここでいう探索の進度とは，初期解と最終的に得られる解との評価値の差を 100 として，初期解からどれだけ評価値が改善されたかを 0 から 100 までの数値で表したものである。これらのデータを入力すると，基本スケジューリングエンジンと連携してその問題を 1 回解き，フェーズ移行の基準とするパラメータ値を決定する。さらに，その問題を解いて各フェーズでのルール適用確率を調整することを繰り返し行い，適切な新しいルール適用確率を決定する。

4.3.4.1 フェーズ移行の基準パラメータの調整方式

入力した問題に対し，全てのルールの適用確率が等しい状態においてフェーズに分割しないでタブーサーチを行い，評価値がどの程度まで改善されるか確認する。このとき，解の遷移回数と各解の評価値を記録しておく。そして，タブーサーチ終了後，その記録に基づいて各フェーズの境界となる探索の進度 Prg_q に達した時点における次の 3 種類の値を求める。

- 評価値: VT_q
- 進度が Prg_{q-1} から Prg_q になるまでに要した解の遷移回数: IT_q
- 直前 10 回の解の遷移における評価値改善量の平均値: MT_q

未知の問題に対する計画の作成においては，第 4.3.2.2 節で述べたように IT_q および MT_q をフェーズ移行の基準として使用する。しかし，ルール適用確率の調整にはこれと同じ問題を用いるので，その際にはより簡単に VT_q を使用する。

4.3.4.2 ルール適用確率の調整方式

基本的には改善能力が高いルールの適用確率が大きくなるように調整するが，中には他のルールの効果を高める形で間接的に解を改善するルールも存在する。そのようなルールの適用確率が小さくなり過ぎると，解を直接改善する

ルールの効果が小さくなってしまう。そうならないようバランスよく調整するのがルール適用確率の調整機能である。次に示す2つのステップを繰り返し，最後に最も良い解が得られたときの適用確率を出力する。

(1) 改善能力が高いルールの適用確率を大きくする

(2) 問題を解いて解の評価値が改善されるか確認し，改善されていなければランダムに選んだルールの適用確率を元に戻す

ただし，全てのフェーズでのルール適用確率の調整を一度に通して行うと，各フェーズでの開始時点での解が毎回変化するためにルールの効果の比較が平等に行えず適切な調整結果が得られない可能性がある。そこで，前のフェーズの探索結果に影響を受けないようにするために，各フェーズの調整開始時の解は毎回同じもの，すなわち，第4.3.4節においてフェーズ移行の基準パラメータを決定する際に求めた各フェーズ移行時の解を使用する。

ステップ(1)のアルゴリズムを説明する。ルールが全部でU個あり，ルールr（$1 \leq r \leq U$）の調整前の適用確率をH_r，改善能力の相対的な高さをp_rとする。まず，各ルールの改善能力を測定するために，基本スケジューリングエンジンによりルール適用確率H_r（$1 \leq r \leq U$）の下でサンプル問題を解いてルール毎に次に示す2つの値を測定する。そして，これら2つの積を，平均値0，$-\delta$以上δ以下（$0 < \delta \leq 1$）の範囲に正規化した値をp_r（$1 \leq r \leq U$）とし，$(H_r + p_r)$をルールrの新たな適用確率とすることを考える。δは調整1回あたりの適用確率変化幅の最大値であり，あらかじめシステムに与えておく。

- ルールrの適用1回あたりの平均の評価値改善量
- ルールrの適用により評価値が改善された回数の割合

このとき，全てのルールについて$0 \leq (H_r + p_r) \leq 1$でなければ適用確率として使用できない。そこで，次のように場合分けしてルールrの新たな適用確率の候補H'_rを決定する。

- $(H_s + p_s) < 0$となるルールsが存在する場合:

$$\begin{cases} H'_r = 0 & (r = s) \\ H'_r = H_r + p_r - (H_s + p_s)/(U-1) & (1 \leq r < s,\ s < r \leq U) \end{cases}$$

- $(H_s + p_s) > 1$ となるルール s が存在する場合:

$$\begin{cases} H'_r = 1 & (r = s) \\ H'_r = 0 & (1 \leq r < s,\ s < r \leq U) \end{cases}$$

- 全てのルールについて $0 \leq (H_r + p_r) \leq 1$ である場合:

$$H'_r = H_r + p_r \qquad (1 \leq r \leq U)$$

次に，ステップ (2) について説明する。ステップ (1) で計算した H'_r は直接的に解を改善する能力が大きいルールほど大きな値になっているが，本節冒頭で述べたように，間接的に解を改善するルールの適用確率が小さくなり過ぎると最終的に得られる解が改善されない。そこで，適用確率 H'_r（$1 \leq r \leq U$）の下でサンプル問題を解いた結果と適用確率調整前の解を比較し，最終的な評価値が改善した場合にのみ H'_r を新たな適用確率として採用する。

評価値が悪化した場合，適用確率が小さくなり過ぎたルールを短時間で特定するのは困難であるため，各ルールについて 1/2 の確率で $H'_r = H_r$ とし，さらに $\sum_{r=1}^{U} H'_r = 1$ となるよう全ルールについて H'_r を一律に増減して，それらを新たな適用確率として採用する。

4.3.5 対話型解修正モジュール

評価関数に取り入れることが困難な評価指標の存在のため，並列スケジューリングモジュールによって作成した計画は，専門家が修正する必要がある。ガントチャートで計画を確認しながら修正要求を繰り返し入力可能とするために，対話型解修正モジュールを開発する。

4.3.5.1 ユーザインタフェース

解修正モジュールは対話的なユーザインタフェースを持つ。図 4.7 に示すように，解修正モジュールのメイン画面にはタブーサーチによって作成した解であるグループ計画のガントチャートをグラフィカルに表示する。また，ガント

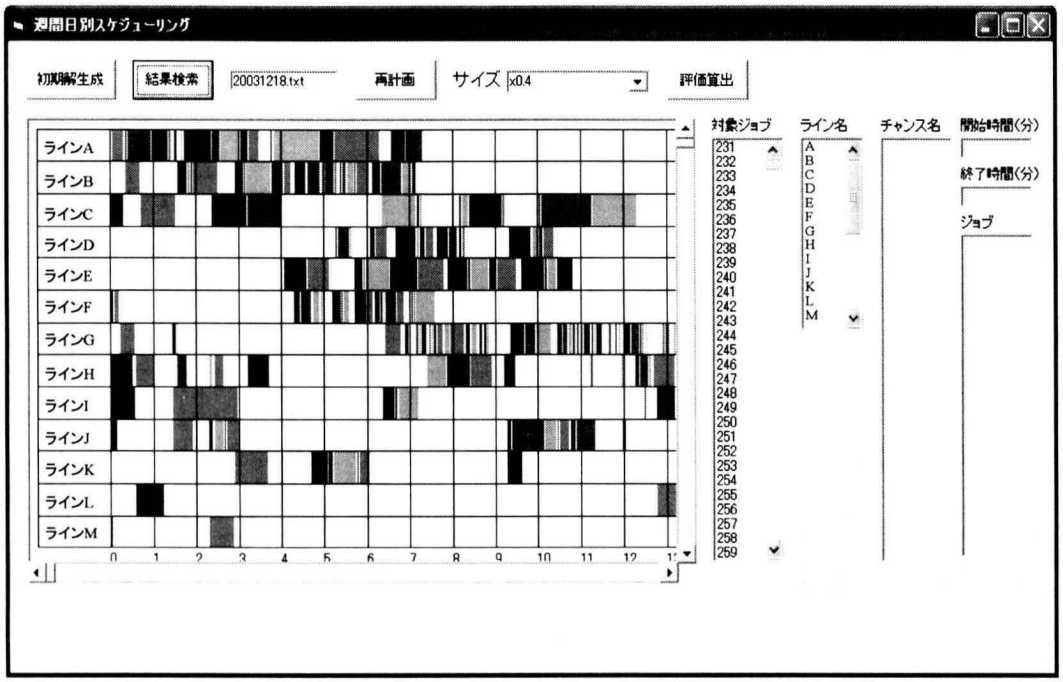

図 4.7: 解修正モジュールのメイン画面

チャートの右側には，その解に含まれるジョブのIDや使用するラインのIDを表示する。

ユーザによって特定のジョブが選択されると，そのジョブを含むグループをガントチャート上で強調表示する。また，特定のラインが選択されると，そのラインで処理するグループの一覧を表示し，そこからグループが選択されると，そのグループの処理開始時刻・終了時刻，グループに含まれるジョブのIDを表示するとともに，ガントチャート上でそのグループを強調表示する。ユーザはこれらの機能によりグループやジョブの詳細な情報を容易に取得することができ，解をどのように修正すべきか判断する材料にできる。メイン画面の評価算出ボタンが押されたら，図4.8に示す評価・制限ダイアログを表示する。評価・制限ダイアログの上段には，修正前の解における評価指標の値を一覧表示する。評価指標には，評価関数の式に含まれる評価指標（平均製造工期，グループ切替回数，ライン休止回数）だけでなく，評価関数に含まれない補助的な評価指標も表示する。評価・制限ダイアログの下段には，解修正の要求として，これらの評価指標に対し特定のラインや全体における最大値または最小値の制限を入力可能とする。入力された制限値をファイルに保存したり，ファイ

評価・制限

評価

ライン名	切替	休止	数	稼働率	最終工期
ラインA	18	0	18	0.997	7260
ラインB	17	3	21	0.925	6168
ラインC	12	0	22	0.998	14759
ラインD	11	2	12	0.341	4093
ラインE	8	1	22	0.769	7955
ラインF	10	2	23	0.649	6044
ラインG	85	3	121	0.924	10433
ラインH	32	7	35	0.713	11752
ラインI	23	0	43	0.973	18287
ラインJ	37	9	38	0.529	14766
総数	389	47	283	平均工期	7052

制限

読み込み　制限更新

ライン名	切替	休止	数	稼働率	最終工期
ラインA	1000	1000	1000	0	100000000
ラインB	1000	1000	1000	0	100000000
ラインC	1000	500	1000	0	100000000
ラインD	1000	1000	1000	0	100000000
ラインE	1000	1000	1000	0	100000000
ラインF	1000	1000	1000	0	100000000
ラインG	1000	1000	1000	0	100000000
ラインH	1000	1000	1000	0	100000000
ラインI	1000	1000	1000	0	100000000
ラインJ	1000	1000	1000	0	100000000
総数	500	10000	10000	平均工期	100000

図 4.8: 解修正の要求の入力画面

ルから制限値を読み込むことも可能である。ユーザが修正の要求を入力後メイン画面に戻って再計画ボタンを押すと，本モジュールは入力された要求を全て満たす解を探索して表示する。修正要求の入力・再計画は，何度でも繰り返すことができる。

4.3.5.2　解修正のアルゴリズム

解修正のアルゴリズムについて説明する。修正要求に用いられる各評価指標に対し，それをタブーサーチによって改善するためのディスパッチングルールを用意しておく。そして，基本スケジューリングエンジンと連携して要求を全

て充足する解を探索する。

特定の評価指標を改善するためのルールはさまざまな特徴を持っており，他のルールが改善しようとする評価指標に対して悪影響を与えてしまうルールや，トレードオフの関係にあるルールも存在する。そこで，ルールを効果的な順番で適用するように，すなわち，目的外の評価指標に悪影響を及ぼすルールを先に適用し，そのルールが目的とする評価指標が要求を満たした後はそのルールをあまり適用しないように，ルール適用確率を動的に変更する。

そのために，各ルールが目的外の評価指標に与える悪影響の範囲や大きさに基づいて，あらかじめルール毎に重みを設定しておく。重みは，そのルールが目的とする評価指標についての要求を満たしている場合の値と満たしていない場合の値を別々に設定する。そして，重みが大きいほど適用確率が大きくなるよう，解修正時には基本スケジューリングエンジンで解の遷移が発生するたびに解修正モジュールが要求の充足具合を確認して，次の近傍集合を作成するための各ルールの適用確率 $H_r(1 \leq r \leq U)$ を式 (4.4) によって決定し基本スケジューリングエンジンに伝達する。ただし，要求 i に対応するルール i の，要求 i が充足されているときの重みを $W1_i$，充足されていないときの重みを $W2_i$，要求 i が充足されているかどうかを表すフラグを G_i（充足されているとき $G_i = 1$，充足されていないとき $G_i = 0$）とする。

$$H_r = \frac{W1_r \times G_r + W2_r \times (1 - G_r)}{\sum_{i=1}^{U}(W1_i \times G_i + W2_i \times (1 - G_i))} \tag{4.4}$$

いずれかの要求 i が満たされたとき，あるいは満たされなくなったとき，計算に使用されるルール i の重みがフラグ G_i によって切り替わるため，ルール適用確率が動的に変化する。その様子を図 4.9 に示す。

図 4.9 のシステムには，要求 1，要求 2，要求 3 に対応するルール 1，ルール 2，ルール 3 がそれぞれ用意されている。ルール 1 の適用は要求 2 や要求 3 に関係する評価指標に対して大きな悪影響を与えるので，要求 1 を満足していない場合にはルール 1 を重点的に使用し，要求 1 を満足した後はルール 1 を使用しないようにするために，ルール 1 の重みは $W1_1 = 0$，$W2_1 = \infty$ に設定してある。また，ルール 2，ルール 3 は互いに影響を及ぼしあうため，対応する要求を満足した後も適用確率が 0 とならないような重みの設定となっている。

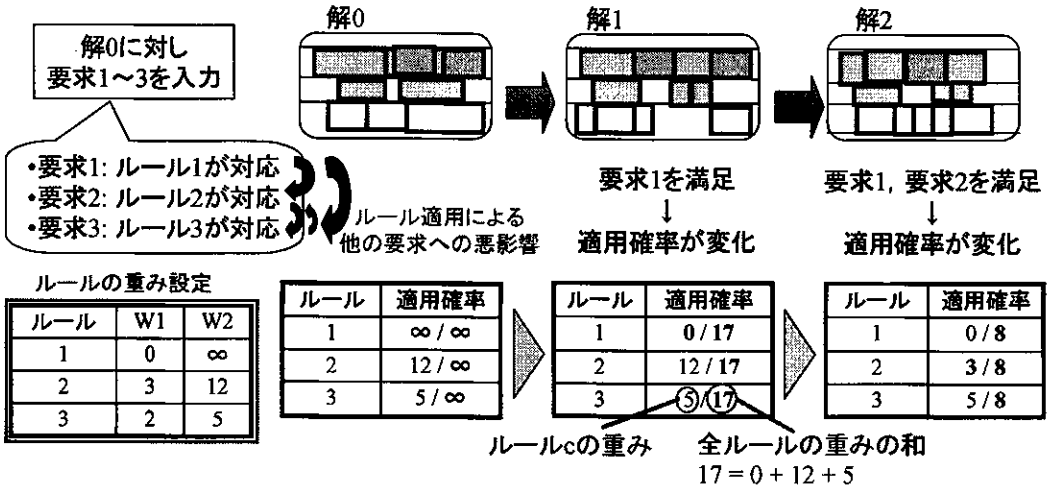

図 4.9: ルール適用確率の動的変化

初めに，解 0 に対して 3 つの要求 1〜3 が入力されたとする。解 0 はどの要求も満足していないので，ルール適用確率の決定に使用する重みは全て $W2$ である。その結果，ルール 1〜3 の適用確率はそれぞれ $H_1 = \infty/\infty = 1$，$H_2 = 12/\infty = 0$，$H_3 = 5/\infty = 0$ となり，要求 1 を満足する解 1 が作成される。要求 1 を満足すると，ルール適用確率の決定に使用するルール 1 の重みが $W1_1 = 0$ に切り替わり，$H_1 = 0$，$H_2 = 12/17$，$H_3 = 5/17$ となる。さらに，要求 2 も満足するとルール 2 の重みもまた $W2_2 = 12$ から $W1_2 = 3$ に切り替わり，未だ満足していない要求 3 に対応するルール 3 の適用確率が高くなる。

4.4 評価実験

4.4.1 実験対象とパラメータ

提案した統合スケジューリングシステムを，グループ制約を持つ大規模な JSP に適用した。実験には，ジョブ数が約 2000，生産ライン数が 13 の，複数の問題を用いた。これらは実際の企業における典型的な生産計画問題であり，その企業においては専門家が手作業により 8 時間かけて計画を作成している。

計算機は，マスタ計算機 1 台（OS: Windows 2000，CPU: Pentium IV 2.0GHz，MEMORY: 512 MB）とタイプ 1 のスレーブ計算機（OS: Windows 2000，CPU: Pentium IV 2.4GHz，MEMORY:512 MB）を複数，およびタイプ 2 のスレー

ブ計算機（マスタ計算機と同スペック）を複数使用した。

(4.1) 式で示した評価関数に含まれる重みは，専門家の経験に基づいて $W_T = 1$，$W_C = 5$，$W_R = 1$ とした。また，提案手法において使用するパラメータには次のものがある。

- 通常のタブーリストのサイズ: k
- ジョブ自由度を構成する各項目の重み: W_{f1}，W_{f2}，W_{f3}
- 初期解の数: $2S$
- 漸近探索タブーサーチにおけるグループ切り替え回数の増加に対する閾値: γ
- フェーズ分割数: F
- 並列タブーサーチにおいて一時的に拡張されたときのタブーリストのサイズ: k_{EX}
- 並列タブーサーチにおいてスレーブでの探索結果を集計するサイクル（解の遷移回数）: E

タブーリストのサイズについては，直前の計画に戻ることを避けるだけで十分な効果があり，反対に大き過ぎると遷移先が限定され過ぎてしまい悪影響が出ることが分かっている。そのため，$k = 1$ とする。初期解の数は，本実験で使用する計算機は高々10 台であるので，$2S = 10$ とする。その他のパラメータについては，以下のように実験によって決定した。

4.4.1.1 ジョブ自由度の各構成要素の重み

(4.3) 式で示したジョブ自由度は，ジョブの必要処理工程数，ジョブが通過するラインの稼働率の和，ジョブの納期の 3 項目から構成され，それぞれ W_{f1}，W_{f2}，W_{f3} という重みを持つ。従来の実験から，W_{f1}，W_{f2} については $W_{f1} = 1000$，$W_{f2} = 100$ とした。W_{f3} については，その値を変化させて作成した初期解に対する並列タブーサーチを 1 時間行い，得られた解の評価値 V を比較した。その結果，表 4.1 に示すように，$W_{f3} = 0.1$ のときに最良の解が得ら

れた。なお，この並列タブーサーチで使用したパラメータは$\gamma = 5$，$k_{EX} = 5$，$E = 10$である。以上より，$W_{f1} = 1000$，$W_{f2} = 100$，$W_{f3} = 0.1$とする。

表 4.1: ジョブ自由度における納期の重みW_{f3}と評価値Vの関係

W_{f3}	0.01	0.1	1
V	13013	12218	12322

4.4.1.2 漸近探索タブーサーチの閾値

漸近探索タブーサーチの有効性およびグループ切り替え回数の増加に対する閾値γの適切な値を確認するために，γの値を1，3，5，7，9，∞と変化させながらそれぞれ1台の計算機で5時間探索する実験を行い，得られた解の評価値Vを比較した。その結果を表4.2に示す。ただし，$\gamma = \infty$は，グループ切替回数の増加に対する制限を設けないことを表す。

$\gamma = 3$のときに最良の評価値が得られている。また，探索時間1時間の場合においても，$\gamma = 3$のときに$V = 13451$，$\gamma = \infty$のときに$V = 14267$であった。このことから，漸近探索タブーサーチは有効であると言える。

以上の実験結果を踏まえ，以降の実験では$\gamma = 3$とする。

表 4.2: 単独でのタブーサーチにおけるパラメータγと評価値Vの関係

γ	0	3	5	7	9	∞
V	16065	12081	13021	12770	13504	13582

4.4.1.3 探索フェーズ分割数

探索フェーズ分割の有効性および適切なフェーズ分割数Fを確認するために，Fの値を1，2，3としてタイプ1のスレーブ計算機3台による1時間の並列タブーサーチをそれぞれ20回ずつ行い，最終的に得られた解の評価値Vの平均値と分散を比較した。ただし，第qフェーズから次のフェーズに移行する探索の進度Prg_qは，経験的にタブーサーチによる改善の傾向が変化しやすいポイントとして，$F = 2$のとき$Prg_1 = 75$，$F = 3$のとき$Prg_1 = 60$，$Prg_2 = 90$と

した。$F=1$ は，フェーズ分割を行わないことを表す。また，ルール適用確率は手作業で調整した値を使用し，並列タブーサーチのパラメータは，$k_{EX}=5$，$E=10$ とした。その結果を表 4.3 に示す。

この結果から，2 つのフェーズに分割したときに，評価値の優れた解が安定して作成されることが分かる。したがって，以降の実験では $F=2$ とする。

表 4.3: フェーズ分割数 F と評価値 V の関係

	V の平均値	V の分散
$F=1$	10981	58763
$F=2$	10697	26413
$F=3$	10939	49660

4.4.1.4 並列タブーサーチのパラメータ

まず，マスタが全スレーブの探索結果を集計するサイクルを $E=10$ とし，スレーブがマスタから解を受信した直後のタブーリストのサイズ k_{EX} の値を変化させながら，1 時間の並列タブーサーチを行った。その結果，表 4.4 に示すように，$k_{EX}=5$ のときに最良の解が得られた。したがって，以降では $k_{EX}=5$ とする。

次に，E の値を変化させながら，1 時間の並列タブーサーチを行った。その結果，表 4.5 に示すように，$E=10$ のときに最良の解を得られた。以上の結果から，以降の実験では $k_{EX}=5$，$E=10$ とする。

表 4.4: パラメータ k_{EX} と評価値 V の関係

k_{EX}	1	5	10
V	12511	12218	12637

表 4.5: パラメータ E と評価値 V の関係

E	5	10	15
V	12683	12218	12589

4.4.2 並列タブーサーチモジュールの評価

4.4.2.1 並列タブーサーチの有効性評価

並列タブーサーチの効率を確認するために，タイプ1のスレーブ計算機3台による並列タブーサーチを行った場合と，同じ計算機3台で解を交換せず独立にタブーサーチを行った場合での解の評価値 V を比較する実験を行った。使用する擬似乱数系列を変えて3回実験を行った結果を表4.6に示す。

並列タブーサーチによる解は，各スレーブが独立にタブーサーチを行ったときの最良解よりも優れている。別のジョブデータに対する実験でも同様の結果となり，並列タブーサーチの優位性が確認できた。

表 4.6: 並列タブーサーチと単独でのタブーサーチの解の評価値

	スレーブ（独立）			並列
	1	2	3	
乱数系列1のときの V	13436	13774	14776	12218
乱数系列2のときの V	13961	13523	14628	12990
乱数系列3のときの V	12826	13589	14819	12200

4.4.2.2 スレーブの台数による並列化効果の変化

スレーブ計算機の数に対して，1時間で得られる解の評価値および専門家による手作業と比べて評価比較95%の解を得られるまでの時間がどのように変化するのか検証した。評価比較（%）は次式で定義される。

$$(\text{評価比較}) = 100/\left(\frac{(\text{提案手法による解の評価値})}{(\text{専門家による解の評価値})}\right)$$

この実験の結果をそれぞれ図4.10，図4.11に示す。ただし，スレーブ計算機にはスペックの異なる2種類のタイプがあるが，高性能なタイプ1の計算機（最大3台）を優先して使用している。

図4.10，図4.11より，スレーブの台数増加による探索効率の向上のしかたは線形ではなく，スレーブ5台でほぼ頭打ちになることが分かる。提案手法においてはスレーブは5台程度で十分であると言える。

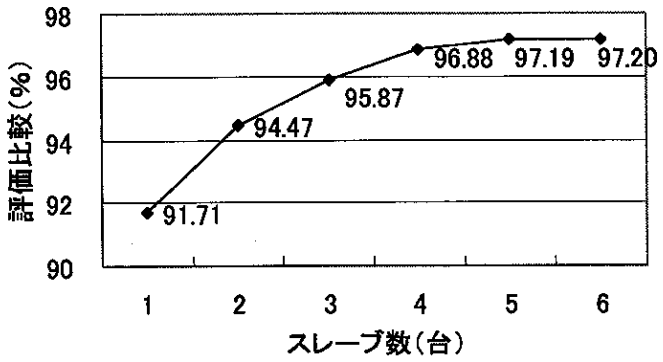

図 4.10: スレーブの台数と解の評価値の関係

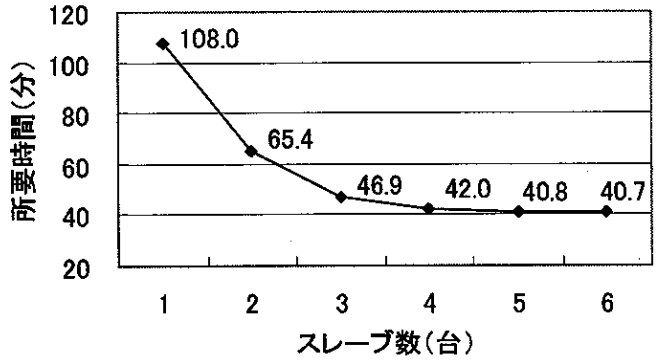

図 4.11: スレーブの台数と探索所要時間の関係

4.4.3 パラメータ自動調整モジュールの評価

提案したパラメータ調整手法を3種類のサンプルデータに適用した。表4.7に，パラメータ調整を進める過程における評価値の平均の変化を示す。また，図4.12に，各ルールの適用確率の変化を示す。なお，適用確率の自動調整においては$\delta = 0.05$とした。このときに要した時間は18時間であり，6回目の調整で得られたパラメータを使用したときに最良の評価値となる解が作成された。

次に，自動調整したルール適用確率を使用する場合と，手作業で50時間かけて調整したルール適用確率を使用する場合について，作成される解の評価値を比較した。調整に使用したのとは別のジョブデータ2種類に対して並列タブー

表 4.7: 自動調整による評価値の変化

調整回数	評価値の平均	調整回数	評価値の平均
1	12619	6	11919
2	12457	7	12156
3	12103	8	12014
4	12276	9	12340
5	12042		

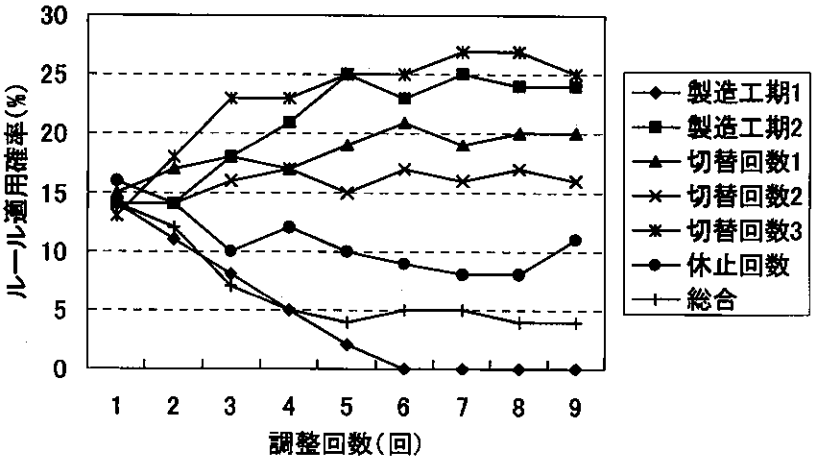

図 4.12: 各ルールの適用確率の変化

表 4.8: ルール適用確率の調整方法による評価値の変化

	データ 1		データ 2	
	評価値の平均	分散	評価値の平均	分散
手作業	10097	50788	13464	79176
提案手法	10128	48468	13013	103400

サーチを 20 回行ったときの，解の評価値の平均と分散を表 4.8 に示す。

表 4.8 から，提案手法により，手作業で調整したルール適用確率を用いた場合と同等の評価値となる解を作成できることが分かる。ルール適用確率を調整

するのに要する時間が手作業では50時間程度であるのに対して提案手法では18時間であり，提案手法が有効であると確認できた。

一方，データ2においては，提案手法によるルール適用確率を用いた場合には手作業の場合と比較して評価値の分散が30%以上大きくなっており，問題によっては評価値の優れた解を安定して得られない場合があることがわかった。

4.4.4 解修正モジュールの評価

入力された要求に応じて解を修正するのに要する計算時間について実験を行った。要求を充足する解を探索する際にルール適用確率を固定した場合と提案手法を適用した場合の，要求が充足されるのに要した時間の比較を図4.13に示す。横軸は同時に入力した要求の数，縦軸はそれらを全て充足する解が作成された時間である。各同時入力数につき，複数のラインに対する要求群（要求群1），同一のラインに対する要求群（要求群2）を用いたところ，全ての場合において提案手法のほうが短時間で修正を終了した。特に，ルール適用確率を固定した場合には要求の同時入力数が増えると指数関数的に計算時間が増加するのに対して，提案手法の計算時間の増加はリニアである。提案手法では5つの要求を同時に入力した場合でも20分程度でそれらを満たす解を作成できており，対話的な解修正を1時間に複数回繰り返すことが可能であることが確認

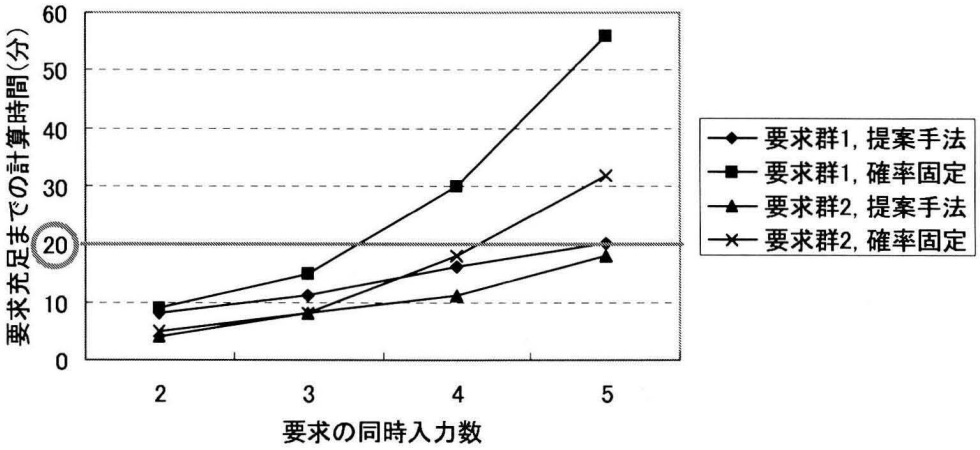

図 4.13: 要求を充足するのに要する計算時間の比較

できた。

また，専門家が望むような計画を提案手法により1時間で作成可能であるか確認するための実験を行った。修正対象とした計画は，図4.7において提案ツールのメイン画面に表示している計画である。この計画の，修正対象となる各評価指標の値を表4.9に示す。

図4.7を見ると，ラインC（上から3行目）やラインH（下から6行目）などに割り当てられた処理は塊が分散しているなど，計画を実行する上で問題となる点がいくつか見て取れる。この解に対し，次に示す要求群X，Y，Zを順次入力して修正した。

- 要求群X
  - ラインF，G，Jの連続処理時間を，それぞれ10時間以上に
  - ラインCの休止回数を1以下に
  - ラインHの休止回数を3以下に
  - ラインKの休止回数を2以下に

表 4.9: 修正前の解の各評価指標

ライン名	切替回数	休止回数	グループ数	稼働率
A	24	0	28	0.997
B	27	3	34	0.861
C	11	2	19	0.781
D	25	3	26	0.512
E	16	1	20	0.477
F	15	2	29	0.605
G	146	3	150	0.580
H	40	8	51	0.619
I	45	3	49	0.296
J	26	6	26	0.309
K	9	4	14	0.377
L	9	2	9	0.231
M	2	1	2	0.339

- 要求群 Y
  - ライン B の切替回数を 25 以下に
  - ライン B の稼働率を 0.9 以上に
  - ライン H のグループ数を 50 以下に
  - ライン H の稼働率を 0.7 以上に
- 要求群 Z
  - ライン B の休止回数を 1 以下に
  - ライン E の稼働率を 0.8 以上に
  - ライン I の稼働率を 0.7 以上に
  - ライン L の切替回数を 5 以下に

各要求群を満足させるのに要した時間は，それぞれ約 12 分，約 21 分，約 15 分であり，合計 48 分という実用的な時間で修正を終えることができた。修正後の解を図 4.14 に，また，その解の各評価指標の値を表 4.10 に，それぞれ示す。

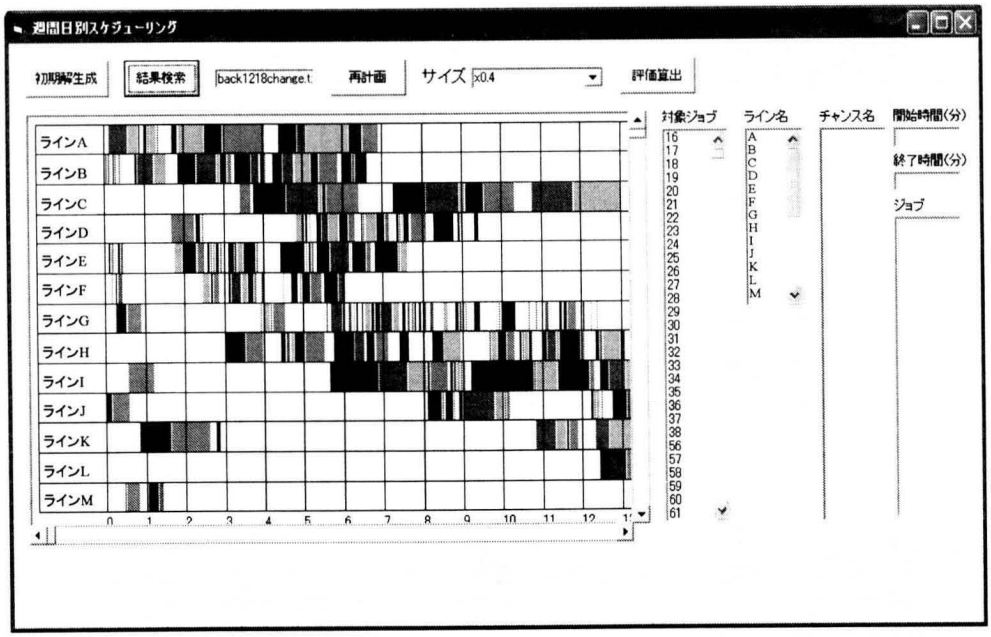

図 4.14: 修正後の解

表 4.10: 修正後の解の各評価指標

ライン名	切替回数	休止回数	グループ数	稼働率
A	23	0	27	0.997
B	24	1	36	0.967
C	7	0	17	0.739
D	32	1	33	0.576
E	16	1	36	0.818
F	17	2	28	0.578
G	137	2	164	0.667
H	34	2	48	0.747
I	32	1	47	0.776
J	20	2	23	0.363
K	12	0	16	0.435
L	4	0	6	0.258
M	2	0	4	0.690

修正の前後で各評価指標を比較すると，グループ数のみ合計 457 から 485 に増加しているが，切替回数の合計が 395 回から 360 回に，休止回数が合計 38 回から 12 回に，稼働率が平均 0.537 から 0.662 に，軒並み改善されている。視覚的にもグループの処理が連続して行われており，各ラインにわたってまとまった計画となっており，良い計画であると言える。

4.5 結言

本章では，グループ制約付き大規模 JSP を対象として，ジョブ数約 2000 の問題において 2 時間以内に専門家が満足する計画を得ることを目標に，確率的ルール適用方式による統合タブーサーチシステムを提案した。

まず，従来の確率的ルール適用方式によるタブーサーチを高速化するために，漸近探索方式および探索フェーズ分割方式を導入し，さらに，複数の計算機を用いる並列化手法を提案した。また，これらの探索手法に用いるパラメータの

自動調整手法を提案した。これらの提案手法により，手作業の場合と比較してパラメータ調整に要する時間を50時間から18時間に短縮でき，また，評価比較95%以上の計画を40分強で作成できた。

そして，評価関数に取り入れることが困難な評価指標に対して計画を改善するための対話型解修正手法を提案した。並列タブーサーチによって得られた評価比較95%の計画を本手法によって修正することにより，50分程度で希望通りの計画を作成できた。これにより，提案システムを用いて専門家と同等の計画を2時間以内に作成可能であることが確認できた。

今後の課題として，ユーザが対話型解修正に時間をかけられるよう，専門家と同等の評価値となる計画の探索時間を短縮することが挙げられる。そのための手段としては，グループ決定に関わる並列タブーサーチの初期解作成手法を改良したり，評価値の優れた計画を安定して得られるようにルール適用確率の調整手法を改良することが考えられる。また，対話型解修正手法において，ルール適用確率を計算するための各ルールの重みの調整を自動化し，効率化・最適化することが望まれる。それから，実際のユーザにヒアリングを行い，ユーザインタフェースを改善することも今後の課題である。

第5章

結論

5.1 本研究のまとめ

本論文では，サプライチェーン（SC）のそれぞれの粒度において効率化を支援することを目的とし，SC全体を対象としたモデリングシミュレーションツール，隣接する2つの企業間で行われる受発注の交渉を自動化するシステム，および，1つの企業内における生産計画問題の代表であるジョブショップ問題（JSP）を対象としたスケジューリングシステムについて論じた。SC全体のモデリング・シミュレーションツールにおいては，SCを機能毎に部分モデルに分割して定義する方式を提案した。自動交渉システムにおいては，問題を制約充足問題としてモデル化し，交渉相手の条件に違反しにくい交渉案を作成して提案しあう方式を提案した。スケジューリングシステムにおいては，従来手法を並列化して高速化する方式，探索パラメータを自動的に調整する方式，対話的に解を修正する方式をそれぞれ提案した。本論文ではこれらの研究成果を以下の4章に分けて述べた。

第1章では，SCの概略とSC効率化の現状について説明した後，SC全体のモデル化・評価，受発注時に行われる交渉，JSPの解法のそれぞれについて関連研究を検討し，本研究での方針を明らかにした。

第2章では，SC全体のシミュレーションを目的としたモデル定義およびその変更を効率的に行うための手法について論じた。シミュレーションの目的がSCの改善であることを考慮して，SCモデルは機能毎に分割して現実に変更しやすい機能を司る部分モデルほど上位となるように階層化し，また，ノードと

アークにより各部分モデルを簡潔に記述可能とした。提案手法を実装して実験を行い，モデル定義に要する時間が汎用ツールの3.6%，シミュレーション結果の実データとの誤差が0.274%であるなど，提案手法の有効性を示した。

第3章では，受発注時に行われる交渉を計算機同士で自動的に行うための手法について論じた。各交渉者が複数の合意の条件を持つ制約充足問題として問題をモデル化し，互いに相手の合意の条件が不明という前提の下，相手が作成した交渉案を自分の条件を満たすよう修正することを繰り返すことにより，両者の条件を同時に満たす合意案を探索する手法を提案した。交渉案の修正時には知識を用いることにより，相手の条件に違反しにくい交渉案を作成可能とした。机上の実験により提案手法を用いて合意案を発見可能であることを確認し，考察を行った。

第4章では，グループ制約を持つ大規模JSPを対象に，専門家が満足する解を高速に作成するための手法について論じた。従来の確率的ルール適用型タブーサーチ手法に漸近探索方式と探索フェーズ分割方式を導入し，また，探索に用いるパラメータを自動調整する手法を提案した。さらに，探索を複数の計算機で並列化することにより，従来の半分の時間で専門家との評価比較95%の解を導出可能とした。また，評価関数化が困難な評価指標について対話的に解を修正する手法を提案し，ガントチャート表示と修正要求入力のインタフェースを持つ統合スケジューリングシステムを実装した。実験により，提案システムによって2時間以内に専門家が満足する解を作成可能であることを確認した。

5.2 今後の研究課題

SCにおいては効率化だけでなく製品を安定して供給することも重要であり，事故や災害などの環境変化に柔軟に対応しながら生産・配送を行うことが必要である。そのような目的のSCMシステムとして著者らは，生産現場における環境変化に対応するために，アトラクター選択を用いたリアルタイム生産スケジューリング方式[86, 87]や環境変化発生時に実行する生産計画をあらかじめ作成してRF-ID（Radio Frequency Identification）タグ[88, 89]に格納し部品に添付するロバスト生産スケジューリング方式[90]，また，一部の道路・倉庫などが事故や災害により使用できなくなった際に柔軟に配送を行うために，製品に添付したRF-IDタグの情報と倉庫–倉庫間・倉庫–物流拠点間の情報交換

に基づき各倉庫が自律的に配送計画を作成するシステム [91] の構想を提案している。これらはまだプロトタイプの段階であるので，実用化するためにより現実的な条件での評価実験を行い，検討・改良を重ねてゆく必要がある。

また，無線通信機能を持つ複数のセンサをマルチホップアドホックネットワーク化した無線センサネットワーク [92, 93] を工場や輸送用車両，倉庫などに構築し，生産現場での機械故障などを未然に発見したり [94]，輸送中・保管中の製品をモニタリングして品質劣化を早期に発見し代品を手配することを可能とするシステムが考えられる。このように，近年目覚しく発達してきている小型の無線情報機器を活用すればSC のサービスレベル向上や低コスト化が期待でき，新しいSCM モデルを構築することが今後の課題である。

謝辞

本研究の全過程を通じて，終始懇切丁寧なるご指導とご鞭撻，格別のご配慮を賜わりました大阪大学大学院情報科学研究科マルチメディア工学専攻 薦田憲久教授に深く感謝申し上げます。

学部，大学院博士前期課程ならびに後期課程において，情報工学全般に関して親切なるご指導とご助言を賜わるとともに，本研究をまとめるにあたり，貴重なお時間を割いて頂き，丁寧なるご教示を賜わりました大阪大学大学院情報科学研究科マルチメディア工学専攻 西尾章治郎教授，岸野文郎教授に深く感謝申し上げます。

本研究の遂行ならびに本研究をまとめるにあたり，直属の指導者として研究の問題点や改良点をご指導頂き，数々のご支援を賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 一階良知助手（現 （株）i-seifu 教育関係事業部長，兼，（学）清風明育社新規校準備室長）に心よりお礼申し上げます。

本研究の遂行ならびに本論文執筆にあたり，貴重なご助言，ご指導を頂きました大阪大学大学院情報科学研究科マルチメディア工学専攻 大川剛直助教授（現 神戸大学大学院自然科学研究科情報・電子科学専攻 教授），秋吉政徳准教授に，心よりお礼申し上げます。

学部，大学院博士前期課程ならびに大学院博士後期課程において，情報工学全般に関して親切なるご指導とご助言を賜りました大阪大学大学院情報科学研究科マルチメディア工学専攻 藤原融教授，サイバーメディアセンター 下條真司教授（現 （独）情報通信研究機構），情報ネットワーク学専攻村上孝三教授，情報システム工学専攻 白川功名誉教授（現 兵庫県立大学大学院応用情報科学研究科長），尾上孝雄教授，藤岡弘名誉教授（現 福井工業大学教授），中前幸治教授，バイオ情報工学専攻 赤澤堅造名誉教授（現 大阪工業大学教授）に深く感謝申し上げます。

本論文の執筆にあたり，勤務時間中に執筆することを快諾してくださり，ま

た，公聴会等に参加するためにご助力を頂きました，松江工業高等専門学校情報工学科 福岡久雄教授，原元司准教授，事務部総務課長 源憲治氏に，心よりお礼申し上げます。

本研究を進める上で多大なるご支援を頂いた（独）日本学術振興会，ならびに文部科学省21世紀COEプログラムに深く感謝申し上げます。

また，第2章の研究に関して，著者が取り組む以前に研究の基礎を築かれ，また，共同研究者としてご助力を頂きましたEindhoven University of TechnologyのJan Goossenaerts氏，薦田研究室の卒業生である圓田哲也氏（現 新キャタピラー三菱（株）），北川智之氏（現 ソニー（株））に深く感謝申し上げます。第3章の研究に関して，著者が取り組む以前に研究の基礎を築かれ，また，共同研究者としてご助力を頂きました薦田研究室の卒業生である長谷川努氏（現 関西電力（株））に深く感謝申し上げます。第4章の研究に関して，グループ制約を持つ大規模ジョブショップ問題に関する多くの資料を快くご提供下さり，度重なる質問にも丁寧にご回答頂き，専門家としての立場からさまざまなご指導，ご助言を頂きました新日鉄ソリューションズ（株）関西支社 池田秀夫氏，長尾佳祐氏，濱本英宜氏，堀内邦義氏，滝井大士氏，今井聡氏に深く感謝申し上げます。また，著者が取り組む以前に研究の基礎を築かれました薦田研究室の卒業生である石原杏平氏（現 パナソニック（株）），共同研究者としてご助力を頂きました大前寿敏氏（現 三洋電機（株）），川村伸幸氏（現 シャープ（株）），河北陽平氏（現 NECシステムテクノロジー（株）），田村慎吾氏（現 関西電力（株）），都子賢志氏（現 （株）日立システムアンドサービス）に深く感謝致します。

同期生として研究室配属から現在に至るまで，研究の進捗においてはさまざまなご助力を下さり，研究以外におきましても親交を頂きました薦田研究室の卒業生である兼田佳和博士（現 （株）日立製作所）に，心よりお礼申し上げます。

研究の日々を通じて様々なご支援を頂いた，秘書の八川三保子氏，杉江真樹氏，中村浩子氏，堀田貴子氏，田端理恵氏および薦田研究室の方々に，深く感謝致します。

最後に，学生時代を通じて温かい支援と叱咤激励をいただいた両親，祖父母，姉弟，研究の合間の息抜きに快く付き合ってくれた友人に深く感謝の意を表します。

参考文献

[1] W. C. Copacino: "*Supply Chain Management: The Basics and Beyond*", The St. Lucie Press (1997).

[2] 同期 ERP 研究所（編）："ERP/サプライチェーン成功の法則", 工業調査会 (1998).

[3] 松原恭司郎（編）："キーワードでわかる SCM・ERP 事典", 日刊工業新聞社 (1999).

[4] ダイヤモンド・ハーバード・ビジネス編集部（編）："サプライチェーン理論と戦略", ダイヤモンド社 (1998).

[5] SCM 研究会（編）："サプライチェーン・マネジメントがわかる本", 日本能率マネジメントセンター (1998).

[6] 井上春樹: "実線サプライチェーン経営革命", 日地出版 (1999).

[7] H. L. Lee and C. Billington: "Managing Supply Chain Inventory: Pitfalls and Opportunities", *Sloan Management Review*, Vol. 33, pp. 65–73 (1992).

[8] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi: "*Design and Managing the Supply Chain — Concepts, Strategies, and Cace Studies*", McGraw-Hill Higher Education (1999).

[9] 日経 BP 社（編）："特集 1 サプライチェーン革命", 日経情報ストラテジー, Vol. 1998/08, pp. 8–37, 日経 BP 社 (1998).

[10] 日経 BP 社（編）："続々と成果を上げるサプライチェーン管理", 日経情報ストラテジー, Vol. 1999/02, pp. 186–191, 日経 BP 社 (1999).

[11] 日経BP社（編）："特集1 徹底検証、サプライチェーンの力学", 日経情報ストラテジー, Vol. 1999/10, pp. 20–32, 日経BP社 (1999).

[12] 杉浦司: "よくわかるITマネジメント", 日本実業出版社 (2000).

[13] 角井亮一: "よくわかるIT物流", 日本実業出版社 (2001).

[14] 薦田憲久, 大川剛直: "システムのモデリングとシミュレーション", 計測自動制御学会 (1995).

[15] 一階良知, 薦田憲久: "サプライチェーンのシミュレーションツール", システム／制御／情報, No. 46, No. 10, pp. 601–606 (2002).

[16] 圓川隆夫, 黒田充, 福田好朗: "生産管理の事典", 朝倉書店 (1999).

[17] 杉山成正（編・著）："～SAP R/3～ ERPによるSCMシステム構築技法", ソフトリサーチセンター (2000).

[18] 中村実, 正田耕一（編・著）："MES入門", 工業調査会 (2000).

[19] 中嶋洋介: "交渉力", 講談社 (2000).

[20] 鈴木光男: "ゲーム理論入門", 共立出版 (1981).

[21] 武藤滋夫: "ゲーム理論入門", 日本経済新聞社 (2001).

[22] A. Rubinstein: "Perfect Equilibrium in a Bargaining Model", *Economica*, Vol. 50, pp. 97–109 (1982).

[23] A. Rubinstein: "A Bargaining Model with Incomplete Information about Time Preferences", *Economica*, Vol. 53, pp. 1151–1172 (1985).

[24] M. Bichler: "A Roadmap to Auction-based Negotiation Protocols for Electronic Commerce", in *Proc. of the 33rd Hawaii International Conference on System Sciences* (in CD-ROM) (2000).

[25] S. Napel: "*Bilateral Bargaining: Theory and Applications*", Springer (2002).

[26] H. Fisher and G. L. Thompson: "Probabilistic Learning Combinations of Local Job-Shop Scheduling Rules", *Industrial Scheduling*, pp. 225–251 (1963).

[27] S. Lawrence: "*Resource Constrained Project Scheduling: an Experimental Investigation of Heuristic Scheduling Techniques (Supplement)*", Graduate School of Industrial Administration, Carnegie-Mellon University (1984).

[28] J. Adamus, E. Balas, and D. Zawack: "The Shifting Bottleneck Procedure for Job Shop Scheduling", *Management Science*, Vol. 34, pp. 391–401 (1988).

[29] D. Applegate and W. Cook: "A Computational Study of the Job-Shop Scheduling Instance", *ORSA Journal on Computing*, Vol. 3, pp. 149–156 (1991).

[30] R. H. Storer, S. D. Wu, and R. Vaccari: "New Search Spaces for Sequencing Instances with Application to Job Shop Scheduling", *Management Science*, Vol. 38, pp. 1495–1509 (1992).

[31] T. Yamada and R. Nakano: "A Genetic Algorithm Applicable to Large-Scale Job-Shop Instances", in R. Manner and B. Manderick (eds.): "*Parallel Instance Solving from Nature 2*", North-Holland, pp. 281–290 (1992).

[32] M. A. A. P. Verwijmeren: "*Networked Inventory Management by Distributed Object Technology*", Koninklijke KPN NV, KPN Research (1998).

[33] D. J. Bowersox and D. J. Closs: "*Logistical Management The Integrated Supply Chain Process*", McGraw-Hill (1996).

[34] H. L. Lee and C. Billington: "Material Management In Decentralized Supply Chains", *Operations Research*, Vol. 41, No. 5, pp. 835–847 (1993).

[35] H. M. H. Hegge: "A Generic Bill-of-Material Processor Using Indirect Identification of Products", *Production Planning & Control*, Vol. 3, No. 3, pp. 336–342 (1992).

[36] "ExSpecT — Executable Specification Tool", http://www.exspect.com/.

[37] K. M. V. Hee: "*Information Systems Engineering: a Formal Approach*", Cambridge University Press (1994).

[38] "Oracle Advanced Supply Chain Planning", http://www.oracle.com/lang/jp/applications/planning/SCMP.html.

[39] C. E. Lindblom and E. J. Woodhouse: "*The Policy-Making Process (Third Edition)*", Prentice Hall (1993).

[40] B. Rubenstein-Montano and R. Malaga: "A Co-Evolutionary Approach to Strategy Design for Decision Makers in Complex Negotiation Situations", in *Proc. of the 33rd Hawaii International Conference on System Sciences* (in CD-ROM) (2000).

[41] 坂和正敏: "非線形システムの最適化", 森北出版 (1986).

[42] 近藤次郎: "オペレーションズ・リサーチ入門 — 計画・管理・運用の技術（NHK ブックス）", 日本放送出版協会 (1978).

[43] 近藤次郎: "オペレーションズ・リサーチ", 岩波書店 (1987).

[44] 森雅夫 他: "オペレーションズリサーチ I", 朝倉書店 (1991).

[45] 今野浩, 鈴木久敏: "整数計画法と組み合せ最適化", 日科技連 (1982).

[46] 茨木俊秀: "組み合せ最適化 — 分枝限定法を中心として", 産業図書 (1983).

[47] R. E. Bellman: "Bounds for the Traveling Salesman Problem", *Operations Research*, Vol. 20, pp. 1044–1056 (1972).

[48] R. E. Bellman（著）, 小田中敏男（訳）: "ダイナミックプログラミング", 東京図書 (1973).

[49] R. E. Bellman and S. E. Dreyfus（著）, 小田中敏男, 有水彊（訳）: "応用ダイナミックプログラミング", 日科技連 (1962).

[50] R. J. White: "*Dynamic Programming*", Oliver and Boyd and Holden-Day (1969).

[51] S. E. Dreyfus and A. M. Law: "*The Art and Theory of Dynamic Programming*", Academic Press (1977).

[52] 電気学会 GA 等組み合せ最適化手法応用調査専門委員会編: "遺伝アルゴリズムとニューラルネット", コロナ社 (1998).

[53] 茨木俊秀: "離散最適化法とアルゴリズム", 岩波書店 (1993).

[54] 田口雄章, 玄光男, 井田憲一: "遺伝的アルゴリズムによる多目的整数計画問題の一解法", 電子情報通信学会論文誌（A）, Vol. 79-A, No. 6, pp. 1221–1223 (1993).

[55] J. H. Holland: "*Adaptation in Natural and Artificial Systems*", The University of Michigan Press (1975).

[56] D. E. Goldberg: "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley (1989).

[57] 森正勝, 玉置久, 荒木光彦: "多目的最適化問題に対する遺伝的アルゴリズムの一構成法", 第 38 回システム制御情報学会研究発表講演予稿集, pp. 341–342 (1994).

[58] M. Languna and F. Glover: "Integrating Target Analysis and Tabu Search for Improved Scheduling Systems", *Expert Systems with Applications*, Vol. 6, pp. 287–297 (1993).

[59] M. Gendreau, P. Soriano, and L. Salvail: "Solving the Maximum Clique Problem Using a Tabu Search Approach", *Annals of Operations Research*, Vol. 41, pp. 385–403 (1993).

[60] F. Glover, E. Taillard, and D. de Werra: "A User's Guide to Tabu Search", *Annals of Operations Research*, Vol. 41, pp. 3–28 (1993).

[61] M. Pinedo: "*Scheduling Theory, Algorithms, and systems*", Prentice Hall (1995).

[62] P. Brucker: "*Scheduling Algorithms, Third Edition*", Springer (2001).

[63] P. V. Bael, D. Devogelaere, and M. Rijckaert: "The Job Shop Solved with Simple, Basic Evolutionary Search Elements", in *Proc. of the Genetic and Evolutionary Computation Conference 1999 (GECCO99)*, pp. 665–669 (1999).

[64] D. C. Mattfeld: "Scalable Search for Scheduling Problems", in *Proc. of the Genetic and Evolutionary Computation Conference 1999 (GECCO99)*, pp. 1616–1621 (1999).

[65] P. V. Bael, D. Devogelaere, and M. Rijckaert: "A Steady-State Evolutionary Algorithm for the Job Shop Problem", in *Proc. of the Genetic and Evolutionary Computation Conference 2000 (GECCO2000)*, pp. 981–986 (2000).

[66] I. Ono, Y. Nagata, and S. Kobayashi: "A Genetic Algorithm Taking Account of Characteristics Preservation for Job Shop Scheduling Problems", in *Proc. of the International Conference on Intelligent Autonomous System 5 (IAS-5)*, pp. 711–718 (2000).

[67] J. Sakuma and S. Kobayashi: "Extrapolation-Directed Crossover for Job-Shop Scheduling Problems: Complementary Combination with JOX", in *Proc. of the Genetic and Evolutionary Computation Conference 2000 (GECCO2000)*, pp. 973–980 (2000).

[68] 池田心, 小林重信: "生得分離モデルを用いた GA と JSP への適用", 人工知能学会論文誌, Vol. 17, No. 5A, pp. 530–538 (2002).

[69] 一階良知, 笹井嵩司, 大川剛直, 薦田憲久: "割当ルール交換可能型最適計画システム", 電気学会論文誌 C（電子・情報・システム部門誌）, Vol. 114, No. 10, pp. 1046–1051 (1994).

[70] J. W. Barnes and J. B. Chambers: "Solving the Job Shop Scheduling Problem with Tabu Search", *IIE Transactions*, Vol. 27, pp. 257–263 (1995).

[71] G. Feitelson and L. Rudolph: "Job Scheduling Strategies for Parallel Processing", in *Proc. of IEEE International Parallel and Distributed Processing Symposium 2000 (IPDPS 2000) Workshop*, pp. 1–17 (2000).

[72] 大前寿敏, 石原杏平, 一階良知, 堀内邦義, 濱本英宜: "グループ制約付き大規模ジョブショップ問題におけるジョブ自由度分析及びタブーサーチによる高速立案方式", 平成15年電気学会電子情報システム部門大会, GS17-3, pp. 1068–1073 (2003).

[73] H. Ohmae, K. Ishihara, Y. Ikkai, K. Horiuchi, and H. Hamamoto: "A High Speed Scheduling Method by Analyzing Job Flexibility and taboo Search for A Large-Scale Job Shop Problem with Group Constraints", in *Proc. of 9th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA03)*, Vol. 2, pp. 227–232 (2003).

[74] 岡宏憲, 一階良知, 薦田憲久: "機能別モデリングによるサプライチェーンシミュレーション方式の流通事例への適用", 電気学会情報システム研究会, IS-02-8, pp. 7–12 (2002).

[75] H. Oka, T. Kitagawa, Y. Ikkai, and N. Komoda: "An Application of a Simulation Tool Using a Multi-Functional Modeling Method to Logistics Field", in *Proc. of IEEE International Symposium on Industrial Electronics 2002 (IEEE-ISIE2002)*, Vol. 4, pp. 1370–1375 (2002).

[76] 岡宏憲, 圓田哲也, 一階良知, 薦田憲久: "サプライチェーンにおける機能別モデル定義方式とシミュレーションツールの開発", 平成14年電気学会電子情報システム部門大会, OS-1-1 (in CD-ROM) (2002).

[77] 岡宏憲, 圓田哲也, 一階良知, 薦田憲久: "サプライチェーンにおける機能別モデル定義方式とそれに基づくシミュレーションツールの開発", 電気学会論文誌C（電子・情報・システム部門誌）, Vol. 123, No. 10, pp. 1884–1891 (2003).

[78] 岡宏憲, 長谷川努, 一階良知, 薦田憲久: "二者間交換分配交渉における目的分析知識を用いた交渉方式", 電気学会情報システム研究会, IS-01-19, pp. 17–22 (2001).

[79] T. Hasegawa, H. Oka, Y. Ikkai, and N. Komoda: "A Negotiation Method Using Improvement Knowledge for Bilateral Exchange and Distribution Negotiation", in *Proc. of 2001 IEEE International Conference on Systems, Man and Cybernetics (SMC'2001)* (in CD-ROM) (2001).

[80] H. Oka, T. Hasegawa, Y. Ikkai, and N. Komoda: "A Bilateral Negotiation Method Using Satisfying Level with Knowledge of Purpose Analysis", in *Proc. of 2001 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2001)*, Vol. 2, pp. 359–364 (2001).

[81] 岡宏憲, 長谷川努, 一階良知, 薦田憲久: "二者間交換分配交渉における目的分析知識を用いた交渉方式", 電気学会論文誌C（電子・情報・システム部門誌）, Vol. 124, No. 1, pp. 189–195 (2004).

[82] Y. Ikkai, H. Oka, and N. Komoda: "A High-Speed Integrated Scheduling System with Tabu Search for Large-Scale Job Shops Problems with Group Constraints", in *Proc. of 5th WSEAS International Conference on Systems Theory and Scientific Computation*, pp. 221–226 (2005).

[83] Y. Ikkai, H. Oka, and N. Komoda: "An Integrated Tabu Search with Probabilistic Rule Application for Large-Scale Job Shop Scheduling Problems with Group Constraints", *WSEAS Transactions on Information Science & Applications*, Issue 12, Vol. 2, pp. 2031–2037 (2005).

[84] 薦田憲久: "機能別モデル記述によるサプライチェーンモデル化とシミュレーションツールに関する研究", JISA会報, No. 66, pp. 93–100 (2002).

[85] J. F. Nash, Jr: "The Bargaining Problem", Econometrica, Vol. 18, No. 2, pp. 155–162 (1950).

[86] H. Chujo, H. Oka, Y. Ikkai, and N. Komoda: "A Real-Time Production Scheduling Method Using Attractor Selection", in *Proc. of 5th International Conference on Computational Intelligence for Modeling Control and Automation (CIMCA'2005)*, Vol. 1, pp. 511–516 (2005).

[87] H. Chujo, H. Oka, Y. Ikkai, and N. Komoda: "A Consideration of Application of Attractor Selection to a Real-Time Production Scheduling", in *Proc. of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology*, pp. 346–354 (2006).

[88] 永田貴士: "無線 IC タグの普及と波及効果", 新規産業レポート, Vol. 52, pp. 1–21, 大和総研 新規産業調査本部 (2006).

[89] 安井潤司, 畠山繁, 柴田貴之, 津田竜哉, 荒木誠一, 福田勝美, 他（著）, NEC 技報編集委員会（編）: "RFID 特集", NEC 技報, Vol. 59, No. 2, pp. 62–123, 日本電気株式会社 (2006).

[90] K. Tuchida, M. Tabuchi, H. Oka, Y. Ikkai, and N. Komoda: "A Robust Scheduling Method for a Job Shop Problem in Production by Using Data Carriers", in *Proc. of 2004 IEEE International Conference on Systems, Man and Cybernetics (SMC2004)* (in CD-ROM) (2004).

[91] Y. Ikkai, H. Oka, N. Komoda, and R. Itsuki: "An Autonomous Distributed Information System for Logistics Control with Data Carriers", in *Proc. of IEEE International Conference on Industrial Technology 2003 (ICIT 2003)*, pp. 1079–1082 (2003).

[92] 山本修一郎: "センサーネットワーク", ビジネスコミュニケーション, Vol. 41, No. 9, pp. 128–133, 株式会社ビジネスコミュニケーション社 (2004).

[93] 末永俊一郎: "無線センサネットワークの適用可能性と課題", UNISYS 技報, Vol. 27, No. 4, pp. 55–64, 日本ユニシス株式会社 (2008).

[94] "センサーネットワーク特集 (前編): ユビキタス JOURNAL | NEC", http://www.nec.co.jp/effort/ubiquitous/2006_0217/.