情報工学実験１　＜最適化アルゴリズムに関する実験＞

担当：内海ゆづ子，生方誠希，本多克宏

目的と概要：知能と情報に関わるさまざまな分野では，システムの最適化やアルゴリズムを用いたシミュレーションに基づく問題解決が試みられる場面が多い．本実験では，知能や情報との関連が深いいくつかの問題に対して，最適化手法や種々のアルゴリズムを用いてシステムの解析やシミュレーションを行うことにより，最適化アルゴリズムに関する理解を深めることを目的とする．また，得られた結果に対する考察の発表を通して，プレゼンテーション能力の向上を目指す．

１．課題の概要

本実験では，実験初日の冒頭で以下のテーマの中からグループごとに指定されるものについて，種々のアルゴリズムを用いたプログラミングを通したシステムの解析やシミュレーションを行う．また，得られた成果に対する考察の発表会を最終日に実施し，発表と質疑応答を通して，最適化アルゴリズムに対する理解を深める．

＜課題一覧＞

- 待ち行列シミュレーション
- 0-1 ナップサック問題
- 巡回セールスマン問題

それぞれの課題については，次ページ以降を参照のこと．また，実験初日に課題が確定した後に，追加資料が配布される場合もある．

２．スケジュール（例年）

本実験では，3 週にわたって，「課題の解説」，「プログラミング」，「発表資料作成」，「成果発表会」を順に行う．その際のスケジュールは以下の通りである．

＜実験スケジュール＞

１コマ目	課題の解説
２コマ目	プログラミング・発表資料作成
３コマ目	プログラミング・発表資料作成
４コマ目	プログラミング・発表資料作成
５コマ目	成果発表会

3．講義形式について

本年度の講義形式は**対面**とする．講義開始時に C5-実習室 1-B, C に集合すること．実習室 B は本実験が，実習室 C は実験 4 で利用する． 班ごとに集まって着席すること． その際， 教卓に向かって左側から若い番号の班が集まるよう， 別紙１に示すように前列に詰めて着席すること．

4．課題の進め方について

プログラミング言語については，何を使用しても良いが，単なるプログラミングの演習ではなく，アルゴリズムの理解・実装から結果に対する考察までを自己解決することが目的であることから，使い慣れた言語で時間内に仕上げることを推奨する．

なお，課題の進行状況に応じて，実験実施時間帯以外にも各自で文献調査や資料作成を行い，成果発表会で充実した発表ができるように準備を進めること．また，成果発表会では，自分の課題だけでなく，他のグループの課題に対しても予習し，質疑応答を通して内容の理解を深めること．

5．予習について

本実験では，3つの課題の中から実験当日に実施課題を指定するが，どの課題が指定されても支障がないように，実験の前日までに，次ページ以降の全ての課題に目を通す，説明資料の動画を視聴するなどして概要を理解してくること．

6. 成果発表会での注意

成果発表会の持ち時間は 1 班あたり 30 分で，20 分発表, 10 分質疑応答である. 時間を厳守すること. また，発表の際は班全員が必ず 1 回は口頭での説明をするよう，担当を決めておくこと.

6. レポート記載・提出の注意

レポートは基本的にガイダンス資料に従って執筆する．但し，グループワークで，どの部分をだれがどれだけ担当したかを示す役割担当を班全員分記載すること.

記載例：

- プログラムの作成
  - 乱数生成（A さん）
  - 待ち行列の実装（B さん）
- 実験
  - パラメータの調整の決定(C さん)
- グラフの描画（D さん）
- スライド作成（A さん XX%, B さん XX%, C さん XX%, D さん XX%）
- 文献の調査（C さん）

レポート提出は授業支援システムを通じて行う.

7. 質問など

各課題に関する質問やコメントは，それぞれの課題の担当者に MS Teams のチャット機能や電子メールで行うこと.

- 内海ゆづ子 yuzuko@omu.ac.jp
- 生方誠希 ubukata@omu.ac.jp
- 本多克宏 khonda@omu.ac.jp

電子メール送信の際には，学籍番号，氏名を必ず記すほか，メールの Subject に [jikken1 question] をつけること.

情報工学実験１＜最適化アルゴリズムに関する実験—待ち行列シミュレーション＞

担当：内海　ゆづ子

１．待ち行列とシミュレーション

図１で示される銀行のATMにおける顧客の流れを例にとる．顧客は開店時間から次々に到来し，ATM機の前に立ち処理を行う（サービスを受けると言う）．このときATMが開いていれば，すぐにサービスを受けることができるが，サービス中の人がいる場合には“待ち”ができ，“行列”が作られることになる．そして，サービス中の顧客は，サービスが終了した人から退出していくという流れを持つ．ATM の設置台数を増やす，あるいは，機器能率を向上させれば，待ち行列は短くなり顧客に対するサービスが向上する．しかしながら，ATM の設置台数を増やしすぎると，“遊び”を持つATMが存在することになり経営効率が下がってしまう．このようなトレードオフの関係のある問題を解析的に解こうというのが待ち行列問題である．

待ち行列のシステムが複雑なとき，あるいは客の到着やサービスの方法が単純でなく解析的な方法がとれないときに，シミュレーションが利用される．シミュレーションを行うためには，現実の事象をモデル化し，シミュレーションし，評価する．そしてその評価結果を基にチューニングを行い再モデル化するという手順が必要となる．

本課題では，図１で示されるように，【設定１．到着過程（顧客がどの位の頻度で到着するか）】と【設定２．サービス過程（顧客が要するサービス時間はどの程度か，ATM数はいくつか）】を分析者がパラメータとして設定し，【解１．待ち行列（どの位の行列ができるのか，顧客は平均どの位の時間待つのか）】と【解２．退出（サービスを受けることができた人数はいくつか）】がどうなるのかをシミュレーションを用いて算出することを目的とする．

－その他の待ち行列問題の対象例を挙げると，車の渋滞（顧客が車，ATM数が道路，サービス時間が法廷速度，待ち行列が渋滞度）といった目に見える行列を持つ問題だけでなく，インターネット回線の混雑（顧客が接続者，ATM数が同時接続可能数，サービス時間が回線利用時間，待ち行列が接続できない人）といった問題にも当てはまる．－

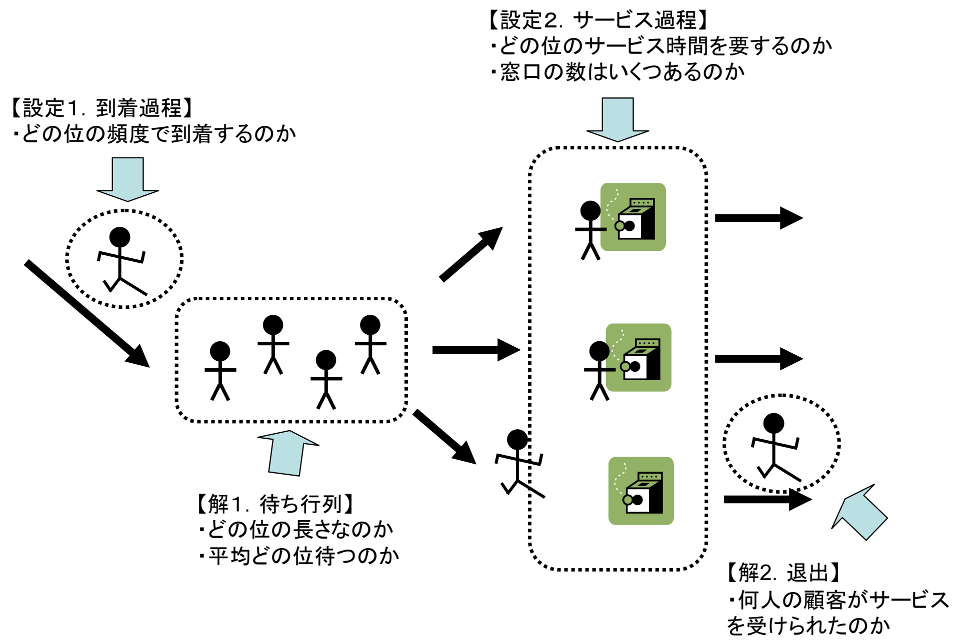


図１　銀行のATMにおける顧客の流れ

２．シミュレーション方法

顧客が到着する，サービスが終了するなどのイベントが発生する時のみ待ち行列は変化するため，イベント発生時の変化のみを考慮する（イベントドリブン型シミュレーション）．待ち行列シミュレーションでは，次の（Ⅰ），（Ⅱ）のイベント発生毎に予定される将来のイベントを考慮する．

（Ⅰ）顧客の到着：次の客の到着時間を予定する．窓口が空いている場合，サービスの終了時刻が予定できる．

（Ⅱ）サービス完了：窓口サービスを完了させ，その窓口を空き状態にする．待ち行列に人がいる場合，次の顧客のサービスが開始されるため，サービス終了時刻が予定できる．

具体的には，予定されたイベントを時刻順に並べておき，時刻をイベント発生予定時刻毎に進めてイベントの実行を行う．決められた時刻まで達したら，シミュレーションを終了する．イベントの管理(スケジューリング)はリスト構造を用いて行えばよい．表１にイベントのスケジューリングとその際の状況の例を示す．

表１　イベントのスケジューリング例

時刻	状態	スケジュール	状態図（窓口が1つの場合）	イベントリスト
0	初期状態	時刻8に顧客1の到着を予定		8: 顧客1の到着
8	顧客1到着	顧客1のサービス開始 時刻14に顧客2の到着を予定 時刻18に顧客1のサービス終了予定	①	14: 顧客2の到着 18: 顧客1のサービス終了
14	顧客2到着	時刻23に顧客3の到着を予定 まだ窓口が処理中のため，サービス終了時刻は未定	①　②	18: 顧客1のサービス終了 23: 顧客3の到着
18	顧客1サービス終了	顧客2のサービス開始 時刻31にサービス終了予定	② ①	23: 顧客3の到着 31: 顧客2のサービス終了
23	顧客3到着	時刻29に顧客4の到着を予定 まだ窓口が処理中のため，サービス終了時刻は未定	②　③	29: 顧客4の到着 31: 顧客2のサービス終了 34: 顧客5の到着
29	顧客4到着	時刻29に顧客4の到着を予定 まだ窓口が処理中のため，サービス終了時刻は未定	②　③④	31: 顧客2のサービス終了 34: 顧客5の到着
31	顧客2サービス終了	顧客3のサービス開始 時刻37にサービス終了予定	③④ ②	34: 顧客5の到着 37: 顧客3のサービス終了

３．顧客の到着時間及びサービス時間の確率分布への当てはめ

【設定１．到着過程】

客がランダムに到着するという仮定を置いた場合は，一般にポアソン分布がよく適用される．単位時間当たりに到着する客数をxとすると，その確率密度関数は

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda} \ (x \geq 0, \lambda > 0) \tag{1}$$

となる．ここでλ は，【平均到着率（人／単位時間）】と呼ばれる指標であり，ポアソン分布の平均と分散は共にλとなる．ここで，平均到着率の逆数$1/\lambda$は【平均到着間隔（時間間隔／人）】という指標になるが，到着間隔の確率密度関数は

$$g(x) = \lambda e^{-\lambda x} \ (x \geq 0, \lambda > 0) \tag{2}$$

という指数分布に従うことがわかっている．なお，指数分布の平均は$1/\lambda$となる．

【設定２．サービス過程】

顧客一人あたりに対するサービスの所要時間をランダムと仮定する．単位時間内に何人にサービスできるかを示す【平均サービス率（人／単位時間）】をμとおく．するとその逆数$1/\mu$は【平均サービス時間（単位時間／人）】という指標になる．当課題では，【平均サービス時間】は指数分布に従うものと考える．

４．指数乱数列の作成方法（逆関数法）

(1),(2) 式より，顧客の【平均到着間隔】と【平均サービス時間】は共に指数分布に従うということがわかった．ここでは，逆関数法を用いて指数分布に従う乱数列の作成方法について述べる．

いま，連続的な確率変数zが確率密度関数$f(x)$を持つとすれば，

$$u = \int_{-\infty}^{z} f(x)dx \tag{3}$$

で定義される確率変数uは，区間$[0,1]$で一様分布することになる．従って一様乱数列$\{u_i\}$に対して

$$u_i = \int_{-\infty}^{z_i} f(x)dx \tag{4}$$

を満たす数列$\{z_i\}$を求めれば，これは確率密度関数$f(x)$に従う乱数列となる．ここで，$f(x)$が指数分布の密度関数であるとすると，

$$f(z) = \lambda e^{-\lambda z} \ (z \geq 0, \lambda > 0) \tag{5}$$

と定式化される．従って（5）式より，指数分布の確率分布関数は，

$$F(z) = \int_{0}^{z} f(x)dx = \int_{0}^{z} \lambda e^{-\lambda x} dx = 1 - e^{-\lambda z} \tag{6}$$

となる．そこで，区間$[0,1]$上の一様乱数$\{u_i\}$からu_nを取り出して

$$u_n = F(z_n) = 1 - e^{-\lambda z_n} \tag{7}$$

を解くと，次の解が得られる．

$$z_n = -\frac{1}{\lambda} \ln(1 - u_n) \tag{8}$$

（8）式は，$[0,1]$の一様乱数列$\{u_i\}$を用いることで，平均$1/\lambda$の指数分布に従う乱数列$\{z_i\}$を求めることができることを示している．

5. 課題

（課題１）銀行 ATM ロビーでの待ち解析１

A 銀行では, 新たに支店を出店することになった. ATM 機を 3 台設置することが決定しており, 一分当たり 2 人の割合での来客が見込まれるものとする. そして, 一人当たりの ATM 利用時間（サービス時間）は平均 2.7 分の指数分布に従うものとする. このとき, 下記の各特性をシミュレーションで解析しなさい.

（ｉ）平均待ち時間
（ⅱ）待ち時間別の顧客頻度数
（ⅲ）待ち行列の長さの時間的な変化状況

（課題２）銀行 ATM ロビーでの待ち解析２

調査結果により, 顧客の来店目的は(a)残高照会, (b)預け入れ, (c)払い出し, (d)振込みの 4 項目に分類できることがわかった. そしてこの来店目的の差異により, ATM 利用時間が異なることも判明している. 下記の表2のようなATM利用目的別のデータが新たに手に入ったものとし, 課題１のモデルの修正を行い解析しなさい.

また, 頻度や所要時間に変化があるときの影響を調査しなさい.

表２: ATM 利用目的別の顧客数

来店目的	(a) 残高照会	(b) 預け入れ	(c) 払い出し	(d) 振込
頻度（人／分）	0.1 (5%)	0.2 (10%)	1.1 (55%)	0.6 (30%)
所要時間（分）	1	2	2	4.5

（課題 3）課題 1, 2 での ATM の台数

課題 1, 2 の条件でパラメータを変えながら実験を行い, 得られた結果をもとに ATM の台数は何台であるべきか考察せよ.

参考文献

「待ち行列」,「オペレーションズ・リサーチ（OR)」,「モンテカルロ・シミュレーション」をキーワードに, 図書館で検索すると良い.

情報工学実験１＜最適化アルゴリズムに関する実験＞

担当：内海　ゆづ子

１．0-1 ナップサック問題（Knapsack problem）

限られた容量を持つナップサックの中に，異なる種類の利得と容量を持つ複数の製品を幾つか詰め込んでいく．このとき，選ばれた製品の容量がナップサックの容量を越えないという条件のもとで，利得の合計を最大化することを考える．

ナップサックの容量がb，製品がn種類あるとし，i番目の製品$(i = 1,2 \cdots, n)$の容量と利得がそれぞれa_i, c_iとして与えられたとする．ここで，決定変数x_i $(i = 1,2,\cdots,n)$を用意しこれらが取り得る値を 0,1 に限定する．そして「$x_i = 0$ならばi番目の製品をナップサックに入れない」，「$x_i = 1$ならばi番目の製品をナップサックに入れる」と解釈すると 0-1 ナップサック問題は以下の(1)式ように整数計画問題として定式化できる．

$$\begin{aligned} &\max \sum_{i=1}^{n} c_i x_i \\ &\mathrm{s.t.} \sum_{i=1}^{n} a_i x_i \ \le b,\ x_i \in \{0,1\} \end{aligned} \tag{1}$$

決定変数x_iが連続値を取る場合，(1)式は線形計画問題となり改定シンプレックス法などのアルゴリズムを利用することで最適解を得ることができる．しかしながら離散値を持つ変数として取り扱う場合には上記最適化アルゴリズムを用いることはできない．

本課題では，(1)式で示される 0-1 ナップサック問題を対象とし，解を得るためのアルゴリズムについての調査と開発を行うことを目的とする．

2. アルゴリズム例

ここでは，既存のアルゴリズムについて 2 つ紹介する．

【完全列挙法】

完全列挙法とは，すべての製品の組み合わせを列挙し，その列挙した組み合わせの中でナップサックの容量制約を満たし，かつ利得が最大となるものピックアップする手法である．全ての組み合わせを網羅するため，必ず最適解が理論的には見つかるという反面，製品の数などの問題数が大きくなると計算量が膨大になるという欠点を持つ．最も単純な方法のため計算手順については省略する．

【欲張り法】

完全列挙法とは異なるヒューリスティックな手法の一つである．価値が高そうな製品から順にナップサックに詰め込んでいく．手軽に解が求まるという反面，その解が最適解とな

る保障がないという欠点を持つ．下記に計算手順を示す．

Step 1. それぞれの製品の評価値を決定しリストを作成する．ここではc_i/a_iとし，少ない容量で大きな利得が得られるものほど価値が高いとする．

Step 2.　リストの中から価値の一番高い製品を選択し，それがナップサックの容量を越えなければナップサックに入れる．越えるのであればリストから削除する．

Step 3. リストに製品がなくなるまで Step 2 を繰り返す．

3. 課題

課題 1

表 1 で示される製品データをもとに，ナップサックの容量が【25】として（1）完全列挙法と（2）欲張り法の 2 通りで解（利得とその組み合わせ）を求めよ．

表１　８つの製品と，各製品に対する容量と利得

製品	1	2	3	4	5	6	7	8
容量	3	5	4	2	10	7	1	5
利得	3	7	6	3	13	9	2	6

課題 2

（1）完全列挙法と（2）欲張り法の違い（長所と短所）を表 1 で示されるような製品データを各自で適宜作成し実験することで考察しなさい．例）製品数を倍の 16 にするなど．

課題 3

整数計画問題を解くアルゴリズムには，その他にも分岐限定法や切除平面法などがある．これらを Web 上から調べ，その特徴について述べなさい．また，応用課題として自分なりのアルゴリズムを開発しなさい．例）完全列挙法と欲張り法を組み合わせる，完全列挙法の一部を省略するなど．

参考文献

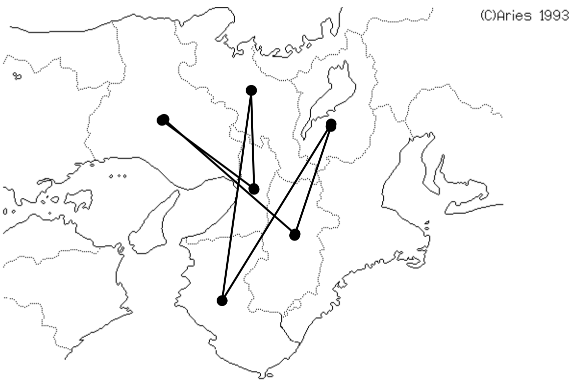
「数理計画法」,「ナップサック問題」,「近似アルゴリズム」をキーワードに検索をするとよい．

情報工学実験１　＜最適化アルゴリズムに関する実験---巡回セールスマン問題＞

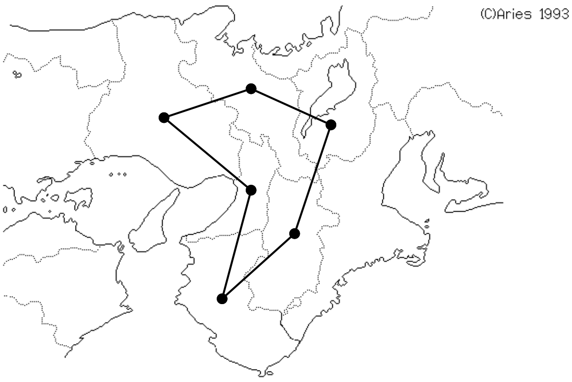
担当：生方 誠希

＜巡回セールスマン問題とは＞

セールスマンの顧客がいくつかの都市に住んでいるとして，ある都市から出発し，他の都市を 1 回ずつ順にすべて訪問して最初の都市に戻ってくる場合に，総移動距離が最小になる巡回路を探索する問題を，巡回セールスマン問題（Traveling Salesman Problem：TSP）と呼ぶ．たとえば，近畿地方の各都市を巡回する場合，



のように巡回するのではなく，



のようにする方が効率的であることは明らかである．このような経路最小化問題は実世界にも存在し，たとえば，コンピュータなどに使われているプリント基板への穴あけや素子配置といった工程の効率化と密接な関係がある．

この最小経路の探索は一見単純に思われるが，実際には計算困難な組み合わせ最適化問題として有名である．いま，都市の数がn個であるとすると，巡回路の総数は$\frac{1}{2}(n-1)!$となるが，仮に$n=30$のときには経路数は4.42×10^{30}にもおよび，そのすべてを調べることは容易でなくなる．このように，問題の大きさに関する多項式に比例した計算時間では厳密な最適解を求めることが困難な問題は NP 困難問題と呼ばれ，種々の工夫を施しながら，なるべく良い解を許容時間内に得ようとするアプローチが必要になる．

本課題では，

(1) 簡便な計算に基づく巡回路の構築法，
(2) 巡回路の改善を繰り返す局所最適解の探索法，
(3) 局所解からの脱出を試みるメタ戦略，
(4) 大域的最適解の探索法

の 4 種類を概説し，シミュレーション実験を通して，計算量と解の精度について考察する．なお，ここでは，2 都市間の距離が向きに無関係な「対称 TSP」を考える．

＜(1) 簡便な巡回路の構築法＞

実行可能な巡回路を直接的に構築する発見的な手法であり，計算が簡便で実装が容易なアルゴリズムが種々，考案されている．以下にそのいくつかを紹介する．

a) 最近隣（Nearest Neighbor）法

任意の都市から出発し，まだ訪問していない都市の中で現在位置から最も近い都市へ移動する方法．これを繰り返し，すべての都市を訪問したら出発点に戻る．

b) 多断片（Multiple Fragment）法

小さい閉路や 3 都市以上とつながる都市が生じない条件の下で，距離最小の 2 都市を順につないでいく方法．これを繰り返し，すべての都市が一つの閉路を構成したら終了する．

c) セービング法

任意の都市から出発し，その時点の巡回路（最初は出発都市のみで構成される）に最も移動距離が増えない位置に都市を一つずつ挿入していく方法．すべての都市が巡回路に組み込まれれば終了する．

＜(2) 局所最適解の探索法＞

前節の方法などにより構築済みの巡回路を，部分的に修正することにより改善するアプローチであり，それ以上の改善が不可能となる解（局所最適解）が得られるまで修正を繰り返す．「初期の巡回路」や「部分的な修正の方法」によっては，必ずしも全巡回路で最も良い解（大域的な最適解）に到達できるとは限らないが，比較的に短時間の操作で前節の解を改善できる点で有効である．「部分的な修正」を行う際に改善の候補を探索する「近傍」の設定方法により，いくつかの手法がある．以下，その代表例を紹介する．

a) 挿入近傍

構築済みの巡回路に対して，任意の 1 都市を順路の別の位置に移動（挿入）することで得られる巡回路を改善の候補とし，候補の中で移動距離の短縮が最も大きいものを次の解とする．そして，更新後の解について同様に改善の候補を作成して，次の候補を探索する．改善が可能な限りこの操作を続け，候補の中に改善される解が無くなれば終了する．

b) 2-opt 近傍

構築済みの巡回路に対して，任意の 2 本の枝を付け替えることで得られる巡回路を改善の候補とし，候補の中で移動距離の短縮が最も大きいものを次の解とする．そして，更新後の解について同様に改善の候補を作成して，次の候補を探索する．改善が可能な限りこの操作を続け，候補の中に改善される解が無くなれば終了する．（3 本の枝を付け替える 3-opt 近傍などもある．）

c) Or-opt 近傍

構築済みの巡回路に対して，任意の部分路を順路の別の位置に移動（挿入）することで得られる巡回路を改善の候補とし，候補の中で移動距離の短縮が最も大きいものを次の解とする．そして，更新後の解について同様に改善の候補を作成して，次の候補を探索する．改善が可能な限りこの操作を続け，候補の中に改善される解が無くなれば終了する．（部分路の長さを 3 都市に固定する場合もある）

＜(3) 局所解からの脱出を試みるメタ戦略＞

前節の局所解探索法では，初期巡回路の取り方によっては大域的な最適解に到達できないため，場合によっては精度の悪い解となっている可能性がある．これは，解の探索が狭い範囲に集中していることに起因している．そこで，局所解からの脱出を試みる工夫を施すことで，なるべく広い範囲を探索しながら最適解に到達しようとするメタ戦略がいくつか提案されている．以下に，いくつかを紹介する．（ここでは，実装が容易なように各戦略を簡略化して記述している．実用の際にはそれぞれ高度な処理が付加される．詳細は，文献を参照のこと．）

a) 多出発（Multiple Start）局所探索

初期巡回路を種々に変えながら局所探索を多数回実行し，その中で最も総移動距離が短いものを解として採用する手法．

b) 擬似焼きなまし（Simulated Annealing）法

局所探索において，解を更新する際に，「候補の中の最良解を採用する」代わりに，「候補から適当に選んだ一つの解が改善すればその解を採用する．改善しない場合は，ある確率pでその解を採用し，それ以外は再び候補から他の解を選ぶ」を繰り返す．すなわち，ある確率で改悪解も採用することで，局所領域からの脱出を試みる．その際，「金属やガラスの焼きなまし（高温から徐々に冷ます工程）」における冷却カーブに似せた減少関数で確率pを変化させることで，探索の初期には改悪も許して広い範囲を探し，後期には局所探索を重視することとなる．

c) 禁断探索（Tabu Search）法

局所探索において局所解（改善する解の候補がない）に到達した際に，候補の中で総移動距離の増加が最も少ない解に移動することで，局所解からの脱出を試みる方法．ただし，このままでは再び元の局所解に戻ってしまうため，直前に動かした都市（または枝）をしばらくの間，タブーリストに記憶し，タブーリストに含まれる都市（または枝）の変更を禁じた状態で解を更新する．このように，後戻りを禁じながら探索を続けることで広い範囲の探索を行う．

＜(4) 大域的最適解の探索法＞

前節のメタ戦略では，なるべく広い範囲を探索することで良い解を得ようとするが，必ずしも大域的最適解（全体の最適解）に到達する保証はない．したがって，真の最適解を得たい場合には，すべての巡回路を列挙して，その中で最適なものを見つける（列挙法という）か，もしくは，緩和問題などを利用して全探索を簡略化する解法（厳密解法という）を用いる必要がある．巡回セールスマン問題には種々の厳密解法が提案されているが，ここでは省略する．詳細については，組み合わせ最適化問題を取り扱う文献を参照されたし．

＜課題＞

担当教員の指示に従い，(1)～(4)について，それぞれ探索法のプログラムを一つずつ作成せよ．2 次元空間上で n 個(都市数:20 程度)の座標点を一様乱数で発生させることで例題を 3 題作成し，計算の所要時間と解の精度の比較を行い，考察せよ．また，都市数 n を種々変化させ，都市数による影響も調査せよ．

＜参考文献＞

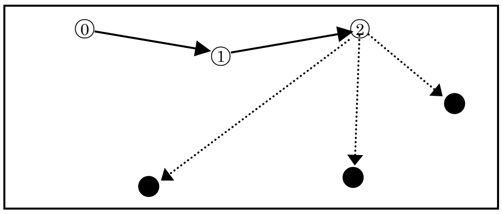
「巡回セールスマン問題」，「オペレーションズリサーチ」，「組合せ最適化」等をキーワードに，インターネットで検索すると良い．

＜最適化アルゴリズムに関する実験---巡回セールスマン問題＞【追加資料】

n 個の都市の座標（x[i], y[i]），i = 0, …, n−1 が与えられた際に，移動距離最小となる巡回路 z[n] = (z[0], …, z[n−1])を求める問題を考える．ただし，z[k]が k 番目に訪問する都市の番号を表すものとする．

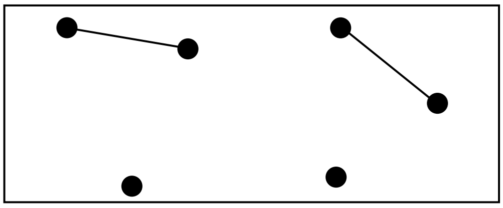
＜(1) 簡便な巡回路の構築法＞

a) 最近隣（Nearest Neighbor）法



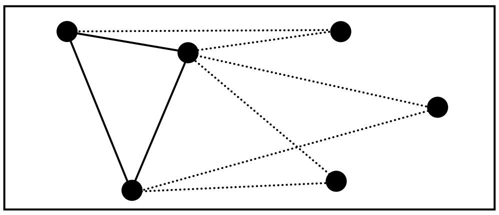
現在の都市から未訪問の他の都市までの距離をすべて求め，最も近い都市を次の訪問都市とする．z[0]を適当に選び，(x[z[0]], y[z[0]])から他の都市までの距離を求めればよい．

b) 多断片（Multiple Fragment）法



すべての都市の間の距離を求め，距離最小の都市から順につないでいく．その際，各都市がどの都市とつながっているかを表す配列 link[n][2] を作成し，都市 i とつながる都市の番号を link[i][0], link[i][1] に記憶する．記憶されている要素の数でつながる都市の数を判定し，順にリンクをたどることで閉路の有無を調べる．

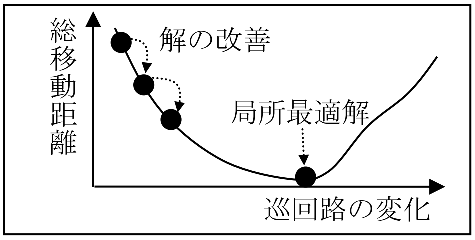
c) セービング法



未訪問の各都市から経路内で最も近い都市を探し，その都市の前か後ろに当該都市を挿入することで新しい巡回路を作成する際に，最も移動距離が増えないものを採用する．

＜(2) 局所最適解の探索法＞

まず，何らかの方法で巡回路（初期巡回路という）が構築済みであるとする．この巡回路の一部分を修正することで，総移動距離を短縮していく．



a) 挿入近傍

構築済みの巡回路の 1 都市を別の場所に移動する．

z[n]の要素

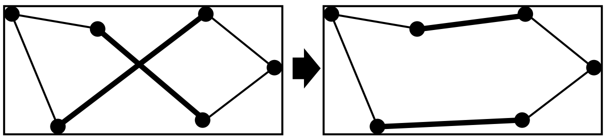
1−**2**−3−4−5−6−1

⇩

1−3−4−**2**−5−6−1

すべての移動の候補（上図の例なら，「1」を固定すれば，「**2**」の移動の候補は 4 箇所ある．これを他の都市についても考える）のなかで，解（総移動距離）が最も改善される解に更新する（z[n]を適宜入れ替える）．いかなる 1 都市の移動でも解が改善されないならば，局所最適解とみなす．

b) 2-opt 近傍



任意の 2 本の枝を付け替えることで解の改善を図る．ここで，2 本の枝の付け替えは，以下のような z[n]の並べ替えに相当する．

z[n]の要素

1−**2−3**−4−**5−6**−1

⇩

1−**2−5**−4−**3−6**−1

すべての枝の組み合わせについて改善されなければ，局所最適解とみなす．

c) Or-opt 近傍

巡回路の部分路（ここでは，都市数を 3 に固定する）を他の場所に移動する．

z[n]の要素

1−$\boxed{\textbf{2−3−4}}$−5−6−1

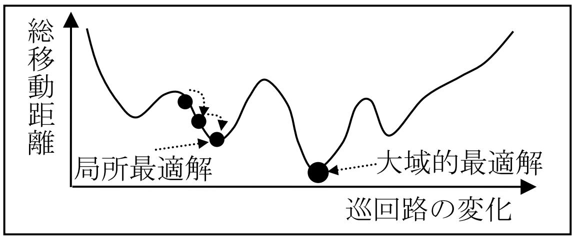
⇩

1−5−**2−3−4**−6−1

すべての候補で改善されなければ，局所最適解とみなす．

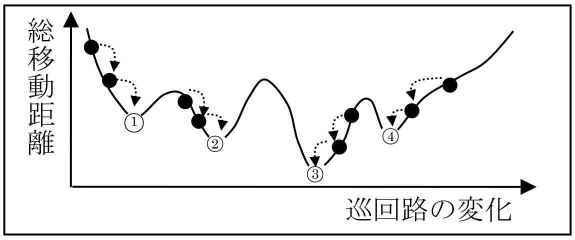
＜(3) 局所解からの脱出を試みるメタ戦略＞

巡回セールスマン問題では，一般に多数の局所解が存在しうる．そこで，局所解からの脱出法を考える．



a) 多出発（Multiple Start）局所探索

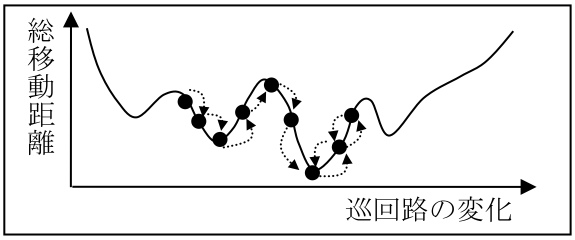
いろんな初期解から探索を行う．



たくさん見つけた局所解の中から一番いいものを採用する．

b) 擬似焼きなまし（Simulated Annealing）法

局所探索を実行する際に，はじめのうちは，改悪される解にも移動することで，局所解からの脱出を試みる．



たとえば，挿入近傍なら，移動する都市と挿入する場所を乱数で適当に定め，その際の総移動距離を求める．解が改善されていれば解を更新．それ以外は，確率 p でその解へ更新し，確率 $(1-p)$ で更新しない．そして，再び移動都市と挿入場所を定めて繰り返す．ただし，$p=0$ なら，通常の局所探索と等しくなる．

改悪解への更新も許容するので，大域的最適解を通り過ぎたり，逆戻りしたりすることもあるが，確率 p を上手く設定することで，p が大きい探索初期には小さな山を乗り越えながら大域的最適解の近くまで進んで行き，p が小さくなる探索後期には最適解に収束する．確率 p を設定する減少関数としては，

$$p = e^{-\Delta/t}$$

などが用いられる．ただし，Δ が評価関数値の増加量を表し，t が焼きなましにおける温度である．t の更新方法（冷却スケジュールと呼ぶ）には，定数 β $(0<\beta<1)$ を用いて繰り返しごとに $t = \beta t$ とおく幾何冷却法や，第 k 反復における温度を，大きな定数 c を用いて $t = \frac{c}{\log(k+1)}$ と定める対数冷却法などがある．いずれにしろ，更新回数が大きくなるほど t，p とも小さくなることから，改悪解への移動割合が減少していく．

探索の終了条件としては，

- あらかじめ定められた反復回数で終了する．
- 温度が十分に低くなったら（ある温度以下になったら）終了する．

などの規則が用いられる．

c) 禁断探索（Tabu Search）法

擬似焼きなまし法と同様に，解の改悪も許しながら大域的最適解を見つけるために，逆戻りを禁じるためのタブーリストを利用する．禁断探索法の枠組みは以下の通りである．

1. 初期巡回路 $\boldsymbol{z}$ を生成する．タブーリスト T を初期化する（空にする）．
2. 局所探索における更新の候補の集合を N とおき，$N \setminus (\boldsymbol{z} \cup T)$ の中で最も総移動距離が短い経路を新たな $\boldsymbol{z}$ とする．$\boldsymbol{z}$ で得られる総移動距離が暫定解よりも短ければ，$\boldsymbol{z}$ を暫定解とする．
3. 終了条件が満たされれば，暫定解を解として出力して終了．それ以外は，T を更新して，2 に戻る．

ここで，タブーリストの作成方法は，用いる局所探索法により種々のアプローチが考えられるが，たとえば，挿入近傍の場合，「ステップ 2 で移動（挿入）した都市をタブーリストに入れ，タブーリスト中の都市を移動（挿入）する巡回路は更新の候補から除外する」という方法がある．また，タブーリストの容量（リストに入りうる都市の数）を設定し，容量に達した際には古いものから順にリストから出す．2-opt 法の場合なら入れ替えた枝を，Or-opt 法の場合なら挿入された枝を，それぞれタブーリストに入れることで，逆戻りを禁じることができる．ただし，タブーリストの容量が「0」ならば，通常の局所探索と等しくなる．

終了条件としては，

- あらかじめ定められた反復回数で終了する．
- あらかじめ定められた反復回数の間に暫定解の更新がなければ終了する．

などの規則が用いられる．

別紙 1

情報工学実験1 実験番号3, 4 着席イメージ

C5 実習室 1-B

実験番号3

教卓

k班	k班		k+1班	k+1班		k+2班	k+2班
k班	k班		k+1班	k+1班		k+2班	k+2班
k班	k班		k+1班	k+1班		k+2班	k+2班

C5 実習室 1-C

実験番号4

教卓

		n班	n班	n+1班	n+1班	n+2班	n+2班
		n班	n班	n+1班	n+1班	n+2班	n+2班
		n班	n班	n+1班	n+1班	n+2班	n+2班