

METRO III (MEta Truck Routing Optimizer Ver. 3.0)
テクニカルドキュメント

Log Opt Co.,Ltd.

平成10年 7月 28日

目次

1	はじめに	5
2	グローバル変数および定数 (Global モジュール)	6
2.1	概要	6
2.2	コード	6
3	ファイル関連ルーチン (File モジュール)	11
3.1	顧客データ読み込みルーチン LoadData_Click	11
3.2	トラックデータ読み込みルーチン LoadTruck_Click	12
3.3	インターデータ読み込みルーチン LoadInter_Click	12
3.4	地図データ読み込みルーチン LoadMap_Click	13
3.5	ルート保存ルーチン SaveRoute_Click	13
3.6	ルート読み込みルーチン LoadRoute_Click	13
3.7	ベンチマーク問題読み込みルーチン Load_Benchmark_Subroutine	13
3.8	コード	15
4	ルート関連基本操作ルーチン (Route モジュール)	27
4.1	次の点を返す関数 R_Next	27
4.1.1	概要	27
4.1.2	コード	27
4.2	前の点を返す関数 R_Prev	27
4.2.1	概要	27
4.2.2	コード	27
4.3	点を別のルートに挿入するルーチン R_Insert	29
4.3.1	概要	29
4.3.2	コード	29
4.4	点をルートから削除するルーチン R_Delete	29
4.4.1	概要	29
4.4.2	コード	29
4.5	ルートの一部を反転するルーチン R_Flip	31
4.5.1	概要	31
4.5.2	コード	31
4.6	ルートの合併を行うルーチン R_Union	32
4.6.1	概要	32
4.6.2	コード	32

5	Graphic 関連ルーチン (Graphic モジュール)	34
5.1	初期化設定ルーチン G.Initialize()	34
5.1.1	概要	34
5.1.2	コード	34
5.2	ルート描画ルーチン G.RouteDraw()	35
5.2.1	概要	35
5.2.2	コード	35
5.3	ノード描画ルーチン G.NodeDraw	37
5.3.1	概要	37
5.3.2	コード	37
5.4	ノード描画ルーチン G.Circle	38
5.4.1	概要	38
5.4.2	コード	38
6	初期解生成 (R.Make モジュール)	39
6.1	ピストン輸送 Piston	39
6.2	コード	39
7	評価値の差の計算ルーチン (Delta モジュール)	41
7.1	点を削除したときの作業時刻の遅れを返す関数 Delta.Delay.Delete	41
7.2	点を挿入したときの作業時刻の遅れを返す関数 Delta.Delay.Insert	42
7.3	点を削除したときの稼働時間超過のペナルティを返す関数 Delta.Time.Delete	42
7.4	点を挿入したときの稼働時間超過のペナルティを返す関数 Delta.Time.Insert	42
7.5	点を削除したときの容量超過のペナルティを返す関数 Delta.Volume.Delete	43
7.6	点を挿入したときの容量超過のペナルティを返す関数 Delta.Volume.Insert	43
7.7	点を削除したときの重量超過のペナルティを返す関数 Delta.Weight.Delete	43
7.8	点を挿入したときの容量超過のペナルティを返す関数 Delta.Weight.Insert	43
7.9	コード	43
8	時間枠関連ルーチン (TW モジュール)	48
8.1	特定のルートの時刻関連の情報を更新するルーチン Time.Compute	48
8.2	全てのルートの時刻関連の情報を更新するルーチン Time.Compute.All	48
8.3	特定のルートの出発時刻を決定するルーチン Departure.Adjust	48
8.4	全てのルートの出発時刻を決定するルーチン Departure.Adjust.All	48
8.5	コード	48
9	タブーサーチ (TS モジュール)	51
9.1	タブーサーチ Tabu.Search3	51
9.2	コード	53
10	ランダム化適応型構築法 (GRASP モジュール)	62
10.1	ランダム化適応型構築法ルーチン GRASP.Subroutine	62
10.2	トラックに種顧客を割り当てるルーチン Set.Seed.Customer	62
10.2.1	入出力	63
10.2.2	アルゴリズム	63
10.3	挿入する顧客を選択するルーチン Select.Best	64

10.4	挿入費用更新ルーチン <code>Compute_Delta(r)</code>	64
10.5	挿入費用初期設定ルーチン <code>Initialize_Delta</code>	64
10.6	<code>a_star</code> だけから成るルートの生成ルーチン <code>Generate_New_Route</code>	64
10.7	まだ挿入されていない顧客リストを更新するルーチン <code>Update_Remain</code>	64
10.8	コード	64
11	最良ルート更新ルーチン (<code>Update</code> モジュール)	71
11.1	最良ルート更新ルーチン <code>BestR_Pool_Update</code>	71
11.2	コード	71
12	ルート選択ヒューリスティック (<code>rsh</code> モジュール)	73
12.1	ルート選択ヒューリスティックルーチン <code>Route_Selection</code>	73
12.2	点のプール内のルールへの挿入費用を計算するルーチン <code>Compute_Delta</code>	73
12.3	コード	73
13	改善法 (<code>OPT</code> モジュール)	75
13.1	ルート改善法 <code>Two_Opt</code>	75
13.1.1	概要	75
13.1.2	コード	75
14	初期値設定 (<code>Default</code> モジュール)	79
14.1	パラメータの初期値設定 <code>DefaultValueSet</code>	79
14.2	コード	79
15	ルート情報表示ルーチン (<code>R_Inform</code> モジュール)	81
15.1	ルート詳細表示ルーチン <code>R_Inform2_Write</code>	81
15.2	ルート詳細表示ルーチン (デバッグ用) <code>R_Inform3_Write</code>	81
15.3	コード	81
16	保存解表示ルーチン (<code>S_Inform</code> モジュール)	90
16.1	保存解表示ルーチン <code>S_Information_Write</code>	90
16.2	コード	90
17	配列初期化ルーチン (<code>Clear</code> モジュール)	92
17.1	配列初期化ルーチン <code>All_Clear</code>	92
17.2	ルート関連配列初期化ルーチン <code>Route_Clear</code>	92
17.3	コード	92
18	その他のルーチン	95
18.1	点を近い順に並びかえるルーチン <code>Near_Make</code> (<code>NearMake</code> モジュール)	95
18.1.1	概要	95
18.1.2	コード	95
18.2	ソートルーチン <code>Quick_Sort</code> (<code>Q_Sort</code> モジュール)	96
18.2.1	概要	96
18.2.2	コード	96
18.3	顧客への費用配分計算ルーチン <code>Cost_Allocation</code> (<code>CostAllo</code> モジュール)	96
18.3.1	概要	96

18.3.2	コード	96
18.4	地点間の費用を計算する関数 Cost (Distance モジュール)	98
18.4.1	概要	98
18.4.2	コード	98
18.5	今後の課題	98

第 1 章

はじめに

ここでは、配送計画ツール METRO III (MEta Truck Routing Optimizer Ver. 3.0) の改良と拡張のために必要なデータ構造および実装（特にアルゴリズム）の詳細について述べる。

一般に、配送計画問題は以下の仮定を持つ問題である。

è デポと呼ばれる特定の地点を出発した運搬車（トラック）が、顧客（需要地点）を経由し再びデポに戻る。

è デポに待機している運搬車の種類および最大積載重量は既知である。

è 顧客の位置は既知であり、各顧客の需要量も事前に与えられている。

è 地点間の移動時間、移動距離、移動費用は既知である。

è 一つのルートに含まれる顧客の需要量の合計は運搬車の最大積載重量を超えない。

è 運搬車の台数は、決められた上限を超えない（ただし、超過した運搬車に対するレンタル料を考える場合もある。）

è 運搬車の稼働時間が一日の上限を超えない（ただし、超過時間を残業費用として考える場合もある。）

現在のバージョンで組み込んでいる付加条件は、以下のものである。

1. 複数種類のトラックの考慮
2. 積載量条件として重量（トン）および容量（ m^3 ）の両者を考慮
3. 一日の稼働時間の上限の考慮（または残業費用の考慮）
4. 時間枠条件（各顧客における作業開始時刻が何時何分から何時何分の間になければならないという条件）の考慮
5. 運搬車と顧客の相性の考慮（例えば、10 トントラックが入れない顧客や、ある種類のトラックでないと訪問できない顧客の考慮）
6. 協力ゲームの理論を用いて総費用を顧客に公平に配分する機能、共同配送のときの費用分担、料金決定などに使用)