Some Details About Meta-Heuristics and Heuristics

Sen

Zhejiang University

March 22, 2022

Contents

Definition

Definition

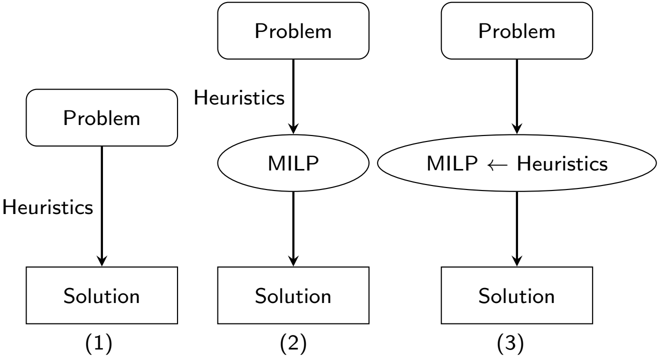
A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithms (Sörensen and Glover, 2013).

The Grand Pattern for Discrete Optimization

$$
\text{Meta-Heuristics} \begin{cases} Single\ State \begin{cases} Local\ Search \\ Tabu\ Search \\ \ldots\ldots \end{cases} \\ Population \begin{cases} Genetic\ Algorithm \\ Particle\ Swarm\ Optimization \\ \ldots\ldots \end{cases} \\ Policy \begin{cases} Q\ Learning \\ \ldots\ldots \end{cases} \\ \ldots\ldots \end{cases}
$$

$$
\text{Heuristics} \begin{cases} MILP \begin{cases} Column\ Generation(10.18) \\ Benders\ Decomposation \\ Cut\ Generation(11.29) \end{cases} \\ Heuristics\ developed\ for\ specific\ problem \end{cases}
$$

Hybrid Approaches



Representation

Processes in Heuristics:

Solution Point $\rightarrow$ Encode $\rightarrow$ Heuristic Functions
Heuristic Functions $\rightarrow$ Decode $\rightarrow$ Solution Point

Type of Representation:

1. Vectors/Lists
   * fixed-length vector
   * queue list
2. Direct Graphs
3. Trees
4. Rule sets

black-and-white travelling salesman problem

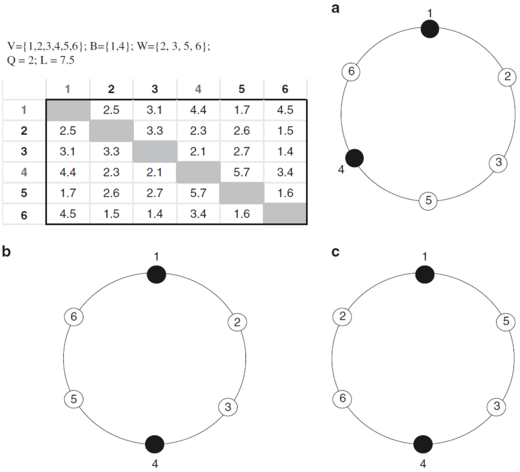
The black-and-white travelling salesman problem(BWTSP) aims to find a shortest Hamiltonian tour(a tour visiting each vertex exactly once) subject to:

i *Cardinality constraint*: the number of white vertices between any two consecutive black vertices may not exceed a positive value Q;

ii *Length constraint*: the distance between any two consecutive black vertices may not exceed a positive value L.

When $Q = L = \infty$, the BWTSP reduces to the TSP. Therefore BWTSP is NP-hard.

Example



V={1,2,3,4,5,6}; B={1,4}; W={2, 3, 5, 6};
Q = 2; L = 7.5

	1	2	3	4	5	6
1		2.5	3.1	4.4	1.7	4.5
2	2.5		3.3	2.3	2.6	1.5
3	3.1	3.3		2.1	2.7	1.4
4	4.4	2.3	2.1		5.7	3.4
5	1.7	2.6	2.7	5.7		1.6
6	4.5	1.5	1.4	3.4	1.6	

Figure: Example

Representation of BWTSP

Fixed-length vector
$(i_0, i_1, i_2, ..., i_{n-1}, i_0)$
How to conduct a local search that satisfies the two constraints?
We need a new solution representation scheme, that captures both the permutation and knapsack features of a BWTSP solution.

Representation of BWTSP

Consider the traditional solution representation, a tour of n vertices $(i_0, i_1, ..., i_{n-1}, i_0)$, where $n = |B| + |W|$.
The tour can be decomposed into $|B|$ paths, each of which starts with a black vertex and extends with ordered white vertices before reaching the next black vertex in the tour. Let $succB(b)$ represent b's immediate successive black vertex in the tour, and $l(b)$ be the length of the path starting with b and ending with $succB(b)$.
Each path starting with b can also be viewed as a knapsack with cardinality kb being the number of white vertices on the path. We name this new representation of a BWTSP solution the knapsack-path structure (KPS).

Representation of BWTSP

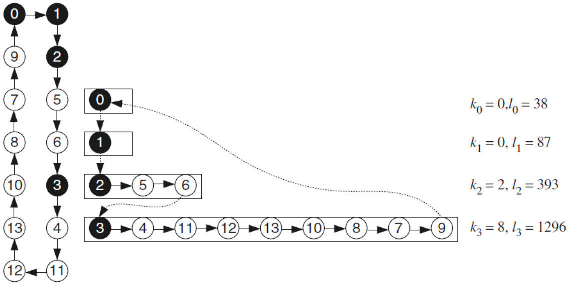


Figure: new solution representation

Local Search

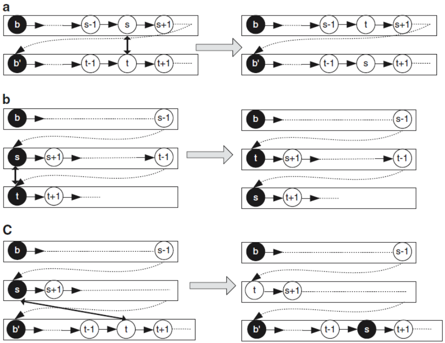


Figure 5 Identifying feasible 2-exchange moves in three cases. **(a)** need to check length feasibility of knapsack paths *b* and *b*'; **(b)** need to check length feasibility of knapsack paths *b, s* and *t*; **(c)** need to check both cardinality and length feasibility of knapsack path *b*.

Figure: Caption

MILP-TS

Algorithm_MILP-TS

1. **Call** *SimpleTS* and output a solution x
2. **If** x is feasible **then**
   Set x as the starting solution for the B&C search tree
   End If
3. Perform (resume) B&C algorithm

3.1 **If** a better incumbent solution x is found **then**
   Use x as an initial solution
   Call *SimpleTS* and output solution x'
   End If

3.2 Set x' as the starting solution for the B&C search tree

3.3 Go to Step 3

End Algorithm

Figure: MILP-TS

The Genetic Algorithm in Lot-sizing Problem

1. What is their point of penetration?
2. What makes their approaches so different?

Model

Lot-sizing

$$\min \sum_{t=1}^{T} (S_t Y_t + C_t X_t + h_t I_t) \tag{1}$$

Subject to,

$$X_t + I_{t-1} - I_t = d_t, \quad \forall t \in T \tag{2}$$

$$X_t \leq M_t Y_t, \quad \forall t \in T \tag{3}$$

$$Y_t \in \{0,\ 1\}, \quad \forall t \in T \tag{4}$$

$$X_t, \quad I_t \geq 0, \quad \forall t \in T \tag{5}$$

Problem

1. Single level OR Multi level
2. Static demand OR Dynamic demand
3. Infinite OR Finite OR Rolling

1. Choice of a representation scheme for a possible solution (coding or chromosome representation.)
2. Decision on how to create the initial population.
3. Definition of the fitness function.
4. Definition of the genetic operators to be used (reproduction, mutation, crossover, elitism)
5. Choice of the parameters of the GAs such as population size, probability of applying genetic operators.
6. Definition of the termination rule.

Inherent defect of GA

(Gen and Cheng 1997).

- In most studies, the roulette wheel selection is used.
- In early generations, the few super chromosomes dominate the selection process.
- In later generations, when the population has largely converged, the selection process degenerate into a random search.
- Sort by fitness function values. Map the function values to some positive real values

peculiarities of the lot sizing problems

(Ozdamar and Birbil (1998))
Crossover operator and repair operation

- The lot sizes of multiple product families and multiple facilities over a planning horizon.
- Use two two square structures of size $J \times T$ to represent a solution.
- Use two-point operator for crossover.
- The crossover process make lead to infeasibility.
- Repair operation.

Order Batch Warehouse

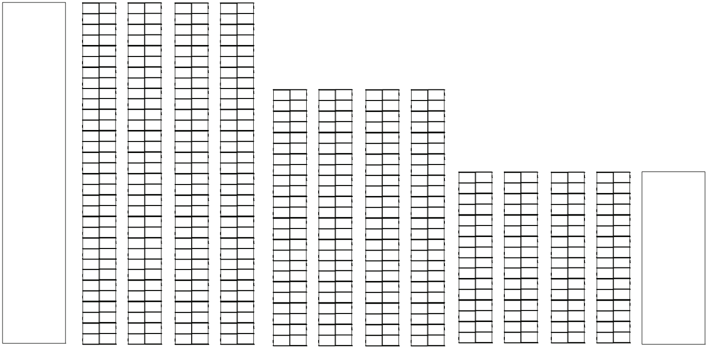


Figure: Warehouse

Model for Return-strategy and the S-shape strategy.

Objective Function

Return

$$\min \alpha \sum_{j \in J} \sum_{k \in K} z_{jk} + (1 - \alpha) \sum_{j \in J} \sum_{b \in B} 2 d_{jb} \tag{6}$$

S-shape

$$\min \alpha \sum_{j \in J} \sum_{k \in K} z_{jk} + (1 - \alpha) \sum_{j \in J} \sum_{b \in B} l_b \delta_{jb} \tag{7}$$

Solution Sample

Random sort generator $\rightarrow$ Sample (1000 points)

$$\rightarrow \text{statistics of } \begin{cases} Heuristicfunctionvalues \\ Originalfunctionvalues \end{cases}$$

Corr

$$\sum_{j \in J} \sum_{b \in B} \delta_{jb}$$

Table: Corr Value 1

	Number of Types	Heuristics Value	S	Return
Number of Types	1	0.082201	0.08884	0.11192
Heuristics Value	0.082201	1	0.92889	0.750911
S	0.08884	0.92889	1	
Return	0.11192	0.750911	0.764585	1

$$\sum_{j\in J}\sum_{b\in B} I_b\delta_{jb}$$

Table: Corr Value 2

	Number of Types	Heuristics Value	S	Return
Number of Types	1	0.123383	0.130444	0.108131
Heuristics Value	0.123383	1	0.999975	0.793467
S	0.130444	0.999975	1	0.79352
Return	0.108131	0.793467	0.79352	1

Abstract Model

$$
\begin{aligned}
\min \quad & cz + dy \\
& Ax \le b \\
& Bx + Gz \le e \\
& Cx + Hy \le f \\
& x, y, z \ge 0 \quad \text{integral}
\end{aligned} \tag{8}
$$

Abstract Model

$$
\begin{aligned}
\min \quad & wy \\
Ax &\leq b \\
Cx + Hy &\leq f \\
x, y &\geq 0 \quad \text{integral}
\end{aligned}
\tag{9}
$$

Abstract Model

$$
\begin{aligned}
\min \quad & w\delta \\
& Ax \leq b \\
Dx + & E\delta \leq g \\
& x, \delta \geq 0 \quad \text{integral}
\end{aligned}
\tag{10}
$$

Abstract Model

The original model

$$\begin{aligned} \min \quad & cx \\ & Ax \le b \\ & x \in Z_+^n \end{aligned} \tag{11}$$

The approximation model

$$\begin{aligned} \min \quad & w\tilde{x} \\ & \tilde{A}\tilde{x} \le \tilde{b} \\ & \tilde{x} \in Z_+^m (m < n) \end{aligned} \tag{12}$$

The End